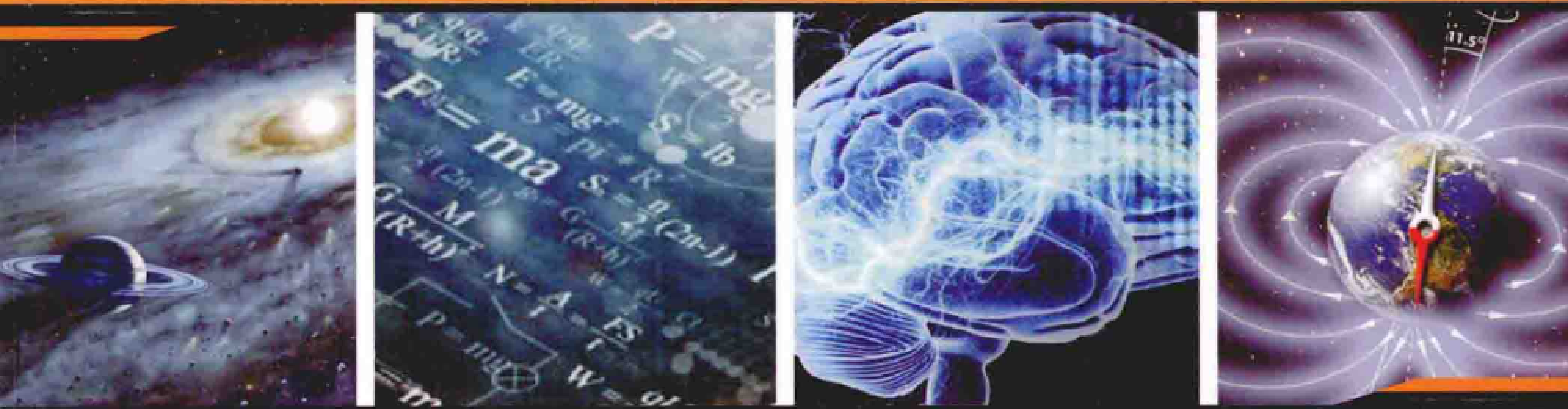
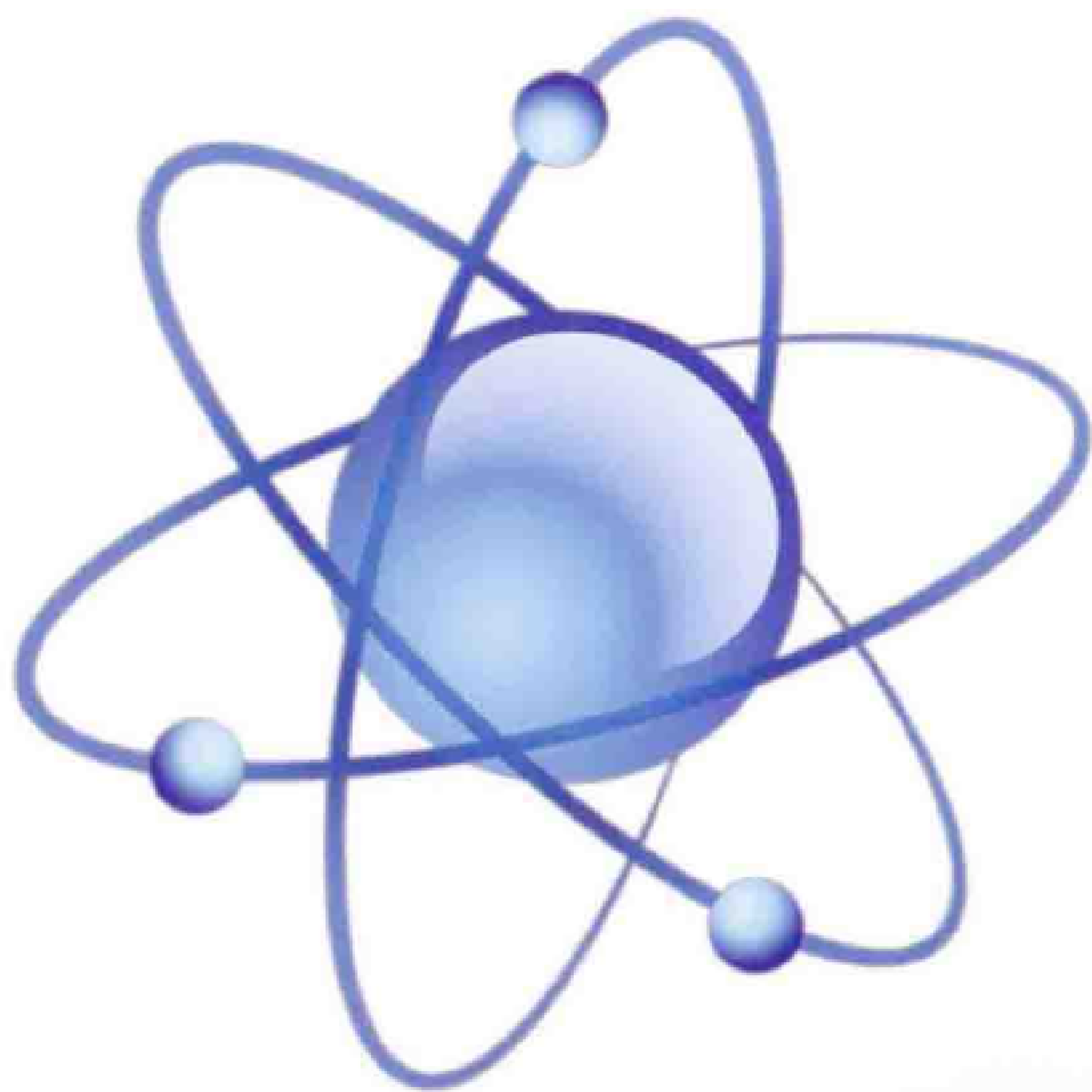
高等学校规划教材·物理学

PLANNING TEXTBOOKS FOR HIGHER EDUCATION



计算物理学基础

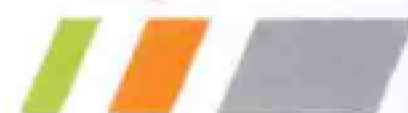
张引科 昝会萍 凌亚文 编著



西北工業大學出版社

高等学校规划教材·物理学

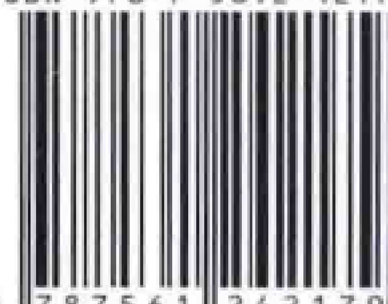
PLANNING TEXTBOOKS FOR HIGHER EDUCATION



□策划编辑／何格夫

□责任编辑／何格夫

□封面设计／扬　帆



ISBN 978-7-5612-4217-9

定价：35.00元

高等学校规划教材·物理学

计算物理学基础

张引科　昝会萍　凌亚文　编著

西北工业大学出版社

【内容简介】 计算物理学是以计算机为工具、以计算技术为手段、运用计算数学方法解决物理问题的应用学科，它与实验物理学、理论物理学一起被并称为现代物理学的三大支柱。本书内容包括常用的数值计算方法及其在物理学中的初步应用。常用的数值计算方法有函数插值与拟合及快速傅里叶变换、数值积分与微分、线性代数方程组的求解、矩阵特征值和特征向量的计算、非线性方程根的求解、常微分方程的解法、解二阶偏微分方程的差分法及蒙特卡罗方法，应用数值计算方法解决物理问题的典型例子有大角度单摆的周期、用单缝衍射方法测量波长、金属电阻温度系数测量、菲涅尔衍射向夫琅禾费衍射的过渡、均匀带电直线段与均匀带电圆环的电场、载流直线段的磁感应强度、直流单臂电桥分析、简单剪切变形的主应变、平行共轴三线圈形成匀强磁场的条件、单摆运动规律分析、扩散现象研究、平行板电容器内部电势的计算、气体自由膨胀与麦克斯韦速率分布的模拟、塞平斯基三角形与羊齿叶图案的绘制等。

对基本计算方法及其在物理学中的应用进行既相对分离又紧密关联的介绍、给出主要计算方法的MATLAB实现程序、有一定数量的物理计算习题是本书的三个突出特点。本书不仅可以作为本科生的教材，还可用作自学计算物理的入门读物，对于科研人员也是具有启发作用的参考书。

图书在版编目(CIP)数据

计算物理学基础/张引科，昝会萍，凌亚文编著．—西安：西北工业大学出版社，2014.12
高等学校规划教材·物理学
ISBN 978-7-5612-4217-9

Ⅰ.①计… Ⅱ.①张… ②昝… ③凌… Ⅲ.①物理学—数值计算—计算方法—高等学校—教材 Ⅳ.①O411

中国版本图书馆CIP数据核字(2014)第299043号

出版发行：西北工业大学出版社
通信地址：西安市友谊西路127号　　邮编：710072
电　　话：(029)88493844　88491757
网　　址：www.nwpup.com
印 刷 者：兴平市博闻印务有限公司
开　　本：787 mm×1 092 mm　1/16
印　　张：16.25
字　　数：388千字
版　　次：2015年1月第1版　　2015年1月第1次印刷
定　　价：35.00元

前　言

物理学是历史最悠久、发展最迅速、应用最广泛、影响最深远的一门基础学科，它对自然科学的其他学科和工程技术的各个领域有着重要的引领作用和巨大的支撑作用。实验研究方法和理论研究方法曾经是物理学普遍采用的两种研究方法，在应用这些方法处理物理问题时，往往会遇到艰深的理论分析、冗长的数据处理或精确的系统控制。在相当多的情况下，这些分析、处理和控制非常艰难或紧迫，人工几乎不可能完成。随着物理系统规模越来越庞大、结构越来越复杂、变化越来越快速或持久，传统的理论研究方法或实验研究方法越来越难以胜任。因此，基于计算机存储量大、运算速度快和控制精度高的优势，研究计算量小、运算速度快、有足够精度的数据处理方法、数值求解方法、数值模拟与仿真方法及实验系统的控制方法就十分必要了，为此计算物理学应运而生。

作为物理学研究的新手段和新方法，计算物理学是对物理学实验研究方法和理论研究方法的补充与拓展。首先，计算物理学可以帮助提高物理实验的自动化程度和实时化水平。计算机的应用使物理实验数据的采集和处理能够同步自动完成，同时还能根据实验中间结果对实验进程进行及时的调整和必要的控制，保证实验向着预定方向进行，提高实验效率。其次，计算物理学能够把物理学家从繁重的公式推导和方程求解中解脱出来。在理论物理学研究中，往往会遇到复杂的公式推演或方程求解，有时甚至不可能取得方程的解析解，影响了对物理规律的探讨和物理本质的认识。利用计算物理学建立的数值计算方法，可以得出物理公式或方程的数值解，并给出相关物理量之间关系的曲线或图像，使物理现象可视化、直观化。再次，计算物理学中的模拟手段和仿真技术使物理学研究更加直观、更加经济。有些物理系统十分复杂，根本无法在实验室重现，有些物理实验耗资巨大，不可能反复进行。对于这些物理过程，人们借助计算物理学方法，模拟和仿真物理系统的变化及物理实验的过程，研究物理系统发展规律或对实验进行优化设计，不仅使不可能的物理研究成为可能，而且能够大幅减少实验成本。

计算物理学成为一门课程开始于20世纪80年代初的美国哈佛大学，80年代中后期我国许多大学的应用物理系开设了这门课程。由于计算物理学是涉及物理学、计算机科学与技术及计算方法的综合性课程，随着计算机的普及、计算机性能的提升及与计算机相关学科的普遍被热捧，加之计算物理学课程对于训练学生思维、培养学生素质、扩展学生应用基础物理知识解决具体实际问题的思路等方面均有裨益，因此计算物理学课程常常能得到学生的欢迎。

在学生对基础学科的兴趣日益淡化但解决技术问题的愿望却逐渐强烈的现实情况下，配合不断强化的素质教育和创新能力培养的新要求，本科学生在掌握了一定的专业基础知识和高等数学知识后，通过学习类似于计算物理学这样的综合应用知识类课程，借助计算机来解决一些单独用理论知识或实验方法都难以解决的问题，能在学习知识的同时享受应用知识的快乐、掌握新方法的快乐及解决新问题的快乐，提升综合素质和创新能力，并由此认识基础知识的重要作用、加深对基础知识的理解。因此，开设计算物理学课程不失为一条素质教育的恰当

途径。

本书的内容主要分为基础物理所涉及的常用数值计算方法和这些方法在具体物理问题中的初步应用两部分。两部分内容既在同一章中分别介绍又相互衔接，使学生在掌握数值计算方法的同时，对其应用有所了解，学习解决问题的思路，以便进一步处理其他问题。另外，计算物理学课程的重要任务之一是编程与上机运行，考虑到计算机的普遍性和编程语言的多样性，本书没有设置编程与上机练习部分，但从两个方面弥补了由此可能带来的不足。一是对主要的计算方法和具体的应用问题都给出了用 MATLAB 软件语言编制的计算程序（可从西北工业大学出版社网站 www.nwpup.com 下载），程序都已经在计算机上运行成功，主要目的是通过这些程序传递计算方法的编程思路。故此，在给出的计算程序中尽量不直接使用 MATLAB 软件的相关函数，也没有单纯追求程序结构的严谨和程序运行的快速，而是把重点放在程序流程的清晰与简洁方面，程序中的详细注释也有助于读者理解程序。二是在每章的习题部分有一定数量的物理问题供读者应用基本数值计算方法来编程解决。编程解决这些问题，既是具有实战性质的上机训练，也是对读者应用知识能力的检验与强化。因此，笔者认为，对基本计算方法及其在物理学中的应用进行既分离又关联的介绍，给出主要问题的结构清晰且能在计算机上顺利运行的 MATLAB 程序，以及在习题部分有一定数量的物理学应用类问题是本书的三个主要特点。另外，笔者没有刻意追求文字叙述的逻辑严谨和公式推导的数学完美，而是把重点放在了对计算方法思路的阐述和应用例子的典型性方面。作为面向本科生的入门级书籍，不可能涵盖计算物理学的方方面面，因此对诸如物理问题的可视化、计算机仿真、实验数据采集与处理、实验设备控制等方面及有限元方法、边界元方法、泛函方法、变分法等计算方法都没有涉及，也没有对 MATLAB 软件进行专门的介绍。

基于以上考虑，本书共分 10 章。内容安排如下：

绪论中介绍了计算物理学的诞生与发展，理论物理学、实验物理学、计算物理学的关系，计算物理学在物理学研究中的作用；

第 1 章介绍了误差的概念及误差的分类，重点论述了舍入误差、有效数字及误差危害的防止措施；

第 2 章的内容有拉格朗日插值法、牛顿插值法和分段低次插值法这三种函数插值方法，函数拟合的最小二乘法及快速傅里叶变换，给出了大角度振动单摆的周期、用单缝衍射方法测量波长、金属电阻温度系数的测量和菲涅尔衍射向夫琅禾费衍射的过渡等四个物理学中的应用问题；

第 3 章介绍了插值型求积公式和差商型数值微分公式、理查森外推法、插值型数值微分公式等三种数值微分方法，这些方法在物理学中应用的例子有两个，一是均匀带电直线段与均匀带电圆环电场的计算，二是载流直线段磁场磁感应强度的计算；

第 4 章包括解线性方程组的直接法、迭代法及范数与方程组的状态，最后对单臂电桥的平衡特性进行了分析；

第 5 章简单叙述了矩阵特征值和特征向量的概念及计算矩阵特征值和特征向量的乘幂法、反幂法及雅可比方法，并分析了简单剪切变形的主应变；

第 6 章讨论了求解非线性方程根的对分法、迭代法、牛顿迭代法和弦截法，以及解非线性方程组的迭代法，讨论了平行共轴三线圈形成匀强磁场的条件；

第 7 章的常微分方程数值解法有欧拉方法、龙格-库塔方法，同时讨论了方法收敛性与稳定性，还有常微分方程组与高阶常微分方程的求解方法，并分三种情况分析了单摆的运动规律；

第 8 章内容有二阶偏微分方程的分类和解的特性、解偏微分方程的差分法，最后分析了扩散现象及平行板电容器内的电势；

第 9 章蒙特卡罗方法的内容主要有随机变量、概率密度与分布函数，随机数的产生，用蒙特卡罗方法计算定积分和求一元方程的实根，蒙特卡罗方法模拟气体的自由膨胀过程和麦克斯韦速率分布规律的建立过程，还对迭代函数系统进行了简单介绍。

在本书的编撰和出版过程中，参阅了大量的书籍和刊物论文，笔者的同事和家人给予了大力的支持与帮助，出版社和本书编辑付出了辛勤劳动，笔者所在单位西安建筑科技大学也提供了大力支持，在此一并向他们表示衷心的谢意。

计算物理学涉及的物理学领域和数值计算方法十分广泛，并且计算物理学的发展也非常迅速，笔者的知识面和能力有限，在本书内容的选择、结构的编排和叙述的方式上还存在不足，敬请各位读者谅解与指正。

编　者

2014 年 7 月

目　录

绪　论

计算物理学(Computational Physics)是以计算机为工具、以计算技术为手段、运用计算数学方法解决物理问题的应用学科,也是利用计算机进行数据采集、数据分析、测量控制和数字仿真来研究物理现象、发现物理规律的交叉学科。计算物理学为研究复杂物理系统的运动规律和结构特性提供了重要手段,对物理学的发展有着巨大的推动作用。近代物理学理论与物理实验技术所取得的重大进展和成果,几乎都是物理学与先进计算机科技密切结合的产物。

1. 计算物理学的诞生与发展

19世纪中叶以前,物理学以实验研究为主,是一门实验科学。1864年,麦克斯韦在对电磁学实验规律总结的基础上,建立了麦克斯韦方程组,创立了电磁理论,随后预言了电磁波的存在并且断言光波就是电磁波,显示了理论研究的巨大潜力。从此理论物理开始成为一个相对独立的物理学分支。20世纪初,量子力学和相对论的诞生,使理论物理学成为较为完整的学科。传统意义上的物理学便有了理论物理学和实验物理学(包括应用物理学)两大支柱,物理学也成为实验和理论密切结合的科学。

物理学与计算机技术及计算方法的结合始于20世纪40年代,计算物理学也诞生于这一时期。在第二次世界大战期间,美国研制核武器时,需要快速准确地处理与热核爆炸有关的大量数据,解决瞬间发生的复杂物理过程的数值计算问题,传统计算方法难以胜任,计算机的应用就成为非常自然的事。之后,随着计算机科学与技术的飞速发展,计算机越来越多地进入物理学研究的各个方面,由此形成了一门独特而丰富的交叉学科——计算物理学。

2. 理论物理学、实验物理学、计算物理学的关系

理论物理学从基本物理原理出发,建立物理问题的方程;用纯数学方法求出方程的解;通过对解所得结论与实验观测结果的对比分析,解释已知现象或发现未知规律,并能为物理实验的设计提供指导。

实验物理学以实验和观测为基本手段来发现新的物理规律、验证物理理论或检验理论研究结论,为理论物理研究的进一步深入奠定基础。

计算物理学是计算机科学与技术、计算方法及物理学三者相互融合的交叉学科,它研究物理学中与数学求解及数据处理等相关的基本计算问题,就是如何以计算机为辅助工具解决物理学问题。

理论物理学、实验物理学、计算物理学之间的关系主要表现在:

(1)实验物理学是理论物理学的基础,也为计算物理学提供所需要的基本数据,同时还能检验理论物理学、计算物理学的研究结果。

(2)理论物理学既可以指导物理实验的设计和实施,又可以为计算物理学提供物理基础。

(3)计算物理学一方面提升了物理学实验的手段,提供了虚拟实验方法,另一方面也为理论物理学研究提供了有力支持。

总之,在对某一复杂物理现象的研究过程中,实验物理学、理论物理学和计算物理学三者

常常相互结合、相互促进。实验物理、理论物理和计算物理是现代物理学的三种主要研究方法。

计算物理学是物理学与计算机技术及数值方法交叉融合的结果。首先,计算物理学以解决物理问题为唯一目的。计算物理学与数值分析不同,它以物理问题为出发点,以揭示物理系统发展规律和变化结果为目标。在用计算物理学处理问题时,只要保持系统的基本物理本质和物理条件不变,就可以利用物理学研究方法直接建立更加简便实用的计算方法,而不必拘泥于严格数值分析理论的限制。其次,计算机技术和数学计算方法的发展,推动了计算物理学的不断进步。计算机存储能力的快速提升和运算速度的持续提高,使物理大系统复杂过程的数值处理成为可能,新的数值计算理论的不断涌现,进一步改善了数值计算的效率和精度,保证了计算物理学研究能力的稳步提升。再次,物理学的进步也为计算机科学与技术的发展构建了坚实的基础。物理学的进展及由此带来的新技术与新材料为计算机科学与技术的突飞猛进提供了理论与物质支撑,物理学研究对计算方法和仿真技术的迫切需求也促进了计算机科学与技术的进步。

3. 计算机在物理学研究中的应用

自 20 世纪 40 年代诞生以来,计算物理学就迅速地向物理学各领域和其他学科渗透。在 20 世纪 60 年代以前,计算机主要用在物理问题的数据计算和简单模拟方面。60 年代以后,计算机进一步用于实验数据的采集与处理、实验过程的控制、理论问题的解析运算、物理过程的计算机模拟与仿真等方面。目前,在物理学研究中,计算机的应用已经是无处不在了。计算机在物理学研究中的应用主要有四个方面:

(1)数值计算与分析。一是在建立描述物理过程的数学公式或方程后,进行数值计算和分析,得出规律性结论,或与实验结果进行对照,或作为下一步实验的基础;二是对实验获得的数据进行分析处理。

(2)解析计算。利用计算机的符号处理系统进行解析计算(包括公式推导和方程求解等)和高精度数值计算,这在理论物理研究中作用重大。

(3)物理过程的模拟与仿真。计算机强大的数值计算功能,为物理学提供了"计算机模拟实验"这种新的研究手段。只要建立了理论模型,就可以进行计算机模拟实验,基本上不受实验条件、时间和空间的限制。

(4)实验过程控制。利用计算机的高速数据采集和处理能力,对物理实验过程进行实时控制。主要包括数据采集、数据分析、设备监控、实验控制等。

物理学是应用计算机较早的学科之一,由此产生的计算物理学及研究方法已经广泛应用到自然科学的其他领域,甚至扩展到包括思维科学、决策科学和管理科学等在内的社会科学领域。

习　　题

1. 什么是计算物理学?
2. 简述实验物理、理论物理和计算物理之间的关系。
3. 简述计算机在物理学研究中应用的几个方面。

第1章　误差及其危害的防止

在计算物理学中，计算机的主要用途就是数值计算工具，研究计算机能够实现的高效数值计算方法是计算物理学的重要任务之一。

人类的计算能力是由计算工具的性能与计算方法的效率共同决定的。从某种意义上来说，采用好的计算方法比提高计算工具的运算速度对改善计算效率更为重要、更为有效、更为经济。据统计，在1955年至1975年的20年中，计算机的运算速度仅提高了几千倍，而解决一定规模椭圆型偏微分方程计算方法的效率竟然提高了100万倍。

数值计算方法是近代数学的一个重要分支，它研究各种数学问题的数值解法（包括数值解法的构造和求解过程的理论分析）以及与此相关的理论（包括计算方法的收敛性、稳定性及误差分析）。数值计算方法又称为计算方法或数值分析，是一门与计算机应用密切结合的、实用性很强的数学课程。

数值计算方法的基本任务是，用计算机能够执行的基本运算，对给定问题的输入数据和计算结果之间的关系给出实际上可行的、理论上可靠的、计算复杂性良好的明确描述。

有的数值计算方法虽然在理论上不够严谨，但通过大量实际计算、对比分析等表明是行之有效的，也可以采用。

数值计算方法给出的解大多是对问题准确解的一种近似，因此误差是数值计算时的一个必须特别重视的问题。若不控制数值计算过程中的误差传递和积累，则计算结果与准确解之间可能会有很大偏差，甚至计算结果完全偏离准确解，使计算失去意义。

本章介绍了误差、相对误差等概念，分析了有效数字与误差的关系，给出了防止误差危害的措施。

1.1　误差及其分类

定义(1.1)　在一定的条件下，物理量必有一个确定的值 x^*（称为客观值或真实值），但这个真实值往往事先并不知道。通常人们采用测量或计算的方法来寻找这个真实值，由于受到测量环境、测量或计算工具及方法等方面多种因素的影响，所找到的值 x 与真实值 x^* 之间总是有一定的差异，这种差异就是误差。即误差是

$$\Delta x = x^* - x \tag{1.1}$$

误差分析包括对误差来源和误差传递规律的分析，并且要对计算或测量结果的误差给出合理估计。

误差的来源是多方面的。根据误差来源的不同，通常把误差分为以下几类：

1. 模型误差

用科学方法解决实际问题时，首先要分析问题、建立模型。模型是对被研究对象的抽象和简化，模型与实际对象之间总是存在一定的差异，不可避免地要带来误差。把应用所选模型进

行研究所得到的准确值与真实值之间的误差称为模型误差。

物理学中的理想模型很多,在这些模型的建立过程中或多或少地进行了简化处理,如忽略了物体的大小后提出了质点模型,不计弹簧质量和物块大小后建立了谐振子模型,不考虑物体变形才有了刚体模型。若以这些理想模型作为基础进行研究,所得到的物理量的准确值必然与该物理量的真实值之间有偏差。这个偏差就是模型误差。

2. 测量误差

对物理量进行测量时,由于测量设备与测量方法的不完善、测量环境的影响、测量人员能力的限制等原因,测量值与真实值之间不可避免地出现误差。这就是测量误差,也称为观测误差。

3. 舍入误差

由于受计算工具(如计算机等)字长或有效数字位数的限制,参与计算的数据只能保留固定位数,多余位则按一定的规则(如"四舍五入")舍掉。这种由于舍掉多余位数字而产生的误差称为舍入误差。

如分别用3.141,1.141,1.732作为π,$\sqrt{2}$,$\sqrt{3}$的近似值时就会带来舍入误差。少量运算次数的舍入误差一般是微不足道的,不会对结果产生明显影响。但是在成千上万次运算过程中,舍入误差可能会被传递并逐渐积累,从而对结果产生显著影响,使计算结果偏离准确值,因此在数值计算时必须对舍入误差给予充分重视。

4. 截断误差

在很多情况下,要得到研究对象数学模型的准确解也是困难的,所以常常用数值方法来求近似解,例如把数学上的无限次计算用有限次计算来近似。这种数学模型的准确解和数值方法的近似解之间的误差称为截断误差。由于截断误差是数值计算方法所固有的,因此又称为方法误差。

如用函数

$$P(x)=x-\frac{1}{3!}x^3$$

来近似函数 $f(x)=\sin x$ 时,就带来截断误差

$$R(x)=f(x)-P(x)=\frac{\cos\xi}{5!}x^5 \quad (\xi\in(0,x))$$

在一般情况下,设函数 $f(x)$ 在点 x_0 的泰勒(Taylor)级数展开公式是

$$f(x)=f(x_0)+f'(x_0)(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1} \tag{1.2}$$

式中,ξ 在 x 与 x_0 之间。记

$$S_n(x)=\sum_{k=0}^{n}\frac{f^{(k)}(x_0)}{k!}(x-x_0)^k \tag{1.3}$$

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1} \tag{1.4}$$

则有

$$f(x)=S_n(x)+R_n(x) \tag{1.5}$$

若把 $S_n(x)$ 作为 $f(x)$ 的近似值，即取

$$f(x) \approx S_n(x) \tag{1.6}$$

则 $f(x)$ 与 $S_n(x)$ 之间的差值 $R_n(x)$ 就是截断误差。即用函数 $f(x)$ 泰勒级数展开的部分和 $S_n(x)$ 来近似函数 $f(x)$ 时，余项 $R_n(x)$ 就是截断误差。

在计算方法中，通常只研究舍入误差和截断误差及其影响。重视误差分析和控制误差的传递与扩散是十分重要的，没有误差分析的数值计算结果是不可信的。

1.2　绝对误差与相对误差

定义(1.2)　设某物理量的准确值为 x^*，若 x 为 x^* 的近似值(如测量结果或数值计算结果)，则称

$$\Delta x = x^* - x \tag{1.7}$$

为近似值 x 的绝对误差，简称误差。

例如 e 取近似值 2.718 2 时，绝对误差为 $\Delta x = \mathrm{e} - 2.718\,2 = 0.000\,081\,8\cdots$。

$|\Delta x|$ 的大小反映了近似值 x 的准确程度。在同一量的几个不同的近似值中，$|\Delta x|$ 越小，近似值 x 的准确度越高。

定义(1.3)　对于实际问题，准确值 x^* 通常是无法确切知道的，从而也无法用式(1.7)计算出近似值的绝对误差，但可以根据问题的背景或计算情况给出误差 Δx 大小的范围。即给出一个正数 ε，使得

$$|\Delta x| = |x^* - x| \leqslant \varepsilon \tag{1.8}$$

成立，则 ε 叫作近似值 x 的绝对误差限，简称误差限，或称为精度。

有了误差限 ε，就能知道准确值 x^* 所在的范围是

$$x - \varepsilon \leqslant x^* \leqslant x + \varepsilon \tag{1.9}$$

这个范围有时也表示成

$$x^* = x \pm \varepsilon \tag{1.10}$$

绝对误差的大小不能恰当地刻画一个近似值的准确程度。如测量 100m 和 1m 两个长度时，若绝对误差都是 1cm，显然前者的测量更为准确。由此可见，要描述一个量近似值的准确程度，除了要考虑近似值绝对误差的大小外，还要考虑该量自身的大小，为此引入相对误差的概念。

定义(1.4)　设 x 为准确值 x^* 的一个近似值，则称

$$e_x = \frac{\Delta x}{x^*} = \frac{x^* - x}{x^*} \tag{1.11}$$

为近似值 x 的相对误差。

一般来说，在同一量或不同量的几个近似值中，相对误差 e_x 的绝对值较小的近似值的精确度较高。由此可见，相对误差比绝对误差更能反映误差的严重程度，因此在误差分析中相对误差比绝对误差更重要。

在实际计算过程中，准确值 x^* 并不知道，但当相对误差 e_x 的绝对值较小时，e_x 表达式(1.11)分母中的准确值 x^* 可以用近似值 x 代替，这种代替产生的影响是 e_x 的高阶无穷小，可以忽略不计。这样就可以用下式计算相对误差：

$$e_x = \frac{\Delta x}{x} = \frac{x^* - x}{x} \tag{1.12}$$

定义(1.5)　由于准确值 x^* 不能事先知道，所以如同绝对误差一样，相对误差也不能准确计算，但可以估计出它的大小范围。即给出一个正数 δ，使得

$$|e_x| = \left|\frac{x^* - x}{x^*}\right| \leqslant \delta \tag{1.13}$$

称 δ 为近似值 x 的相对误差限。

根据式(1.8)和式(1.13)，相对误差限 δ 和绝对误差限 ε 的关系是

$$\delta = \frac{\varepsilon}{|x^*|} \tag{1.14}$$

当相对误差 e_x 的绝对值较小时，相对误差限 δ 满足公式

$$|e_x| = \left|\frac{x^* - x}{x}\right| \leqslant \delta \tag{1.15}$$

绝对误差和绝对误差限有量纲，而相对误差和相对误差限没有量纲，通常用百分数表示。

例 1.1　用千分尺测量一长度所得结果是 $l_0 = 26.267\text{mm}$，又用最小刻度为 mm 的直尺测得该长度的值是 $l = 26.3\text{mm}$，试确定直尺测量结果的绝对误差、绝对误差限、相对误差和相对误差限。

解　千分尺比直尺精度高很多，在计算直尺测量结果的误差时，可以把千分尺的测量结果看作长度的真实值。因此，直尺测量结果的绝对误差是

$$\Delta l = l_0 - l = 26.267 - 26.3 = -0.033\text{mm}$$

直尺的最小刻度为 mm，考虑到测量时的对准与读数时的估计，可以把测量的误差限确定为半个最小刻度，即绝对误差限为

$$\varepsilon = 0.5\text{mm}$$

测量结果的相对误差和相对误差限分别是

$$e_l \approx \frac{l_0 - l}{l} = \frac{26.267 - 26.3}{26.3} \times 100\% = -0.13\%$$

$$\delta \approx \frac{\varepsilon}{l} = \frac{0.5}{26.3} \times 100\% = 1.9\%$$

1.3　有效数字与误差

为了确定用“四舍五入”规则所得近似值的绝对误差限，先看下面的例 1.2。

例 1.2　用“四舍五入”方法分别给出准确值 1.730…，1.731…，1.732…，…，1.739… 保留前三位数字的近似值，并讨论近似值的绝对误差和绝对误差限。

解　各准确值 x_i^*、近似值 x_i、绝对误差 Δx_i 和绝对误差限 ε_i 见表 1.1。

结果表明，用“四舍五入”规则从准确值 1.73… 所确定的近似值为 1.73 或 1.74，它们的共同绝对误差限是 0.005，等于近似值 1.73 或 1.74 的末位数(3 或 4)所在位(小数点后第二位)的单位 0.01 的一半。

可见，凡是由准确值根据“四舍五入”规则得到的近似值，其绝对误差限等于该近似值末位数字所在位单位的一半。

表　1.1

准确值 x_i^*	近似值 x_i	绝对误差 $\Delta x_i = x_i^* - x_i$	绝对误差限 ε_i	绝对误差限统一确定为
1.730…	1.73	0.000…	0.001	$\varepsilon = 0.005 = \frac{1}{2} \times 0.01$
1.731…		0.001…	0.002	
1.732…		0.002…	0.003	
1.733…		0.003…	0.004	
1.734…		0.004…	0.005	
1.735…	1.74	$-0.004\,\underline{\cdots}$[注]	0.005	
1.736…		$-0.003\,\underline{\cdots}$	0.004	
1.737…		$-0.002\,\underline{\cdots}$	0.003	
1.738…		$-0.001\,\underline{\cdots}$	0.002	
1.739…		$-0.000\,\underline{\cdots}$	0.001	

注：$\underline{\cdots} + \cdots = 0.001$。

为了使近似值数字位数的多少能够反映其准确程度，引入有效数字的概念。按照“四舍五入”规则减少数字位数所得近似值的有效数字定义如下：

定义(1.6)　如果按照“四舍五入”规则减少数字位数所得近似值 x 的绝对误差限是某位单位的一半，就称该近似值准确到这一位，并且把从这一位到近似值左边第一位非零数字的全部数字称为这个近似值的有效数字。

例 1.3　设 $a=-2.18$ 和 $b=2.120\,0$ 分别是由准确值经过“四舍五入”得到的近似值，问绝对误差限 ε_a 和 ε_b、相对误差限 δ_a 和 δ_b 各是多少？

解　由准确值经“四舍五入”得到的近似值的绝对误差限等于该近似值末位所在位单位的一半，于是两个近似值的绝对误差限分别是

$$\varepsilon_a = \frac{1}{2} \times 0.01 = 0.005, \quad \varepsilon_b = \frac{1}{2} \times 0.000\,1 = 0.000\,05$$

相对误差限分别是

$$\delta_a = \frac{0.005}{2.18} \approx 0.23\%, \quad \delta_b = \frac{0.000\,05}{2.120\,0} \approx 0.002\,4\%$$

定义(1.7)　设 x 为准确值 x^* 的近似值，将 x 写成

$$x = \pm(x_1 \times 10^{-1} + x_2 \times 10^{-2} + \cdots + x_n \times 10^{-n}) \times 10^m = \pm 0.x_1x_2\cdots x_n \times 10^m \tag{1.16}$$

式中，x_1 是 $1,2,\cdots,9$ 中的数字；$x_2,\cdots,x_n$ 是 $0,1,2,\cdots,9$ 中的数字；n 为正整数；m 是整数。并且近似值 x 的绝对误差满足不等式

$$|x^* - x| \leqslant \frac{1}{2} \times 10^{m-n} \tag{1.17}$$

则称近似值 x 具有 n 位有效数字。

例 1.4　(1) 按“四舍五入”规则写出下列各数据具有五位有效数字的近似值：

187.932 5，　0.037 855 51，　8.000 033

(2) 确定下列各近似值的有效数字位数及绝对误差限：

$$2.000\,4,\quad -0.002\,00,\quad 900\,0$$

解 (1) 对于每一个数,从左边第一个非零数字起向右保留五位数,第六位按"四舍五入"规则舍去或进位,便可得到有五位有效数字的近似值。它们分别是

$$187.93,\quad 0.037\,856,\quad 8.000\,0$$

(2) 近似值从左边第一位非零数字到右边最后一位数字的总位数就是这个近似值的有效数字位数。这几个近似值的有效数字位数分别是5,3,4。近似值的绝对误差限等于末位的半个单位。它们分别是

$$\frac{1}{2}\times 0.000\,1=0.000\,05,\quad \frac{1}{2}\times 0.000\,01=0.000\,005,\quad \frac{1}{2}\times 1=0.5$$

对于表示为式(1.16)形式的近似值,在 m 相同的情况下,n 越大,有效数字的位数越多,10^{m-n} 越小,绝对误差限$10^{m-n}/2$ 就越小,相对误差限也越小。因此,在进行数值计算时,应尽量避免参加运算的数的有效数字位数损失,以免影响计算精度。另外,准确值的有效数字位数可以看作无限多位。

有效数字与误差的关系由以下两个定理给出:

定理(1.1) 若近似值 x 具有式(1.16)的形式,则其相对误差限为

$$\delta_x=\frac{1}{2x_1}\times 10^{-n+1} \tag{1.18}$$

证明 显然

$$x_1\times 10^{m-1}\leqslant |x|\leqslant (x_1+1)\times 10^{m-1}$$

由相对误差的定义式(1.12),得

$$e_x=\left|\frac{x^*-x}{x}\right|\leqslant\frac{10^{m-n}/2}{x_1\times 10^{m-1}}=\frac{1}{2x_1}\times 10^{-n+1}$$

则相对误差限为$(1/2x_1)\times 10^{-n+1}$。

由此可见,只要知道了近似值 x 的有效数字位数 n 和第一位非零数字 x_1,就能估算出它的相对误差限。反之,还可以从近似值的相对误差限来估算其有效数字位数。

定理(1.2) 若近似值 $x=\pm(x_1\times 10^{-1}+x_2\times 10^{-2}+\cdots+x_n\times 10^{-n})\times 10^m$ 的相对误差限为

$$\delta_x=\frac{1}{2(x_1+1)}\times 10^{-n+1} \tag{1.19}$$

则 x 至少具有 n 位有效数字。

证明 由于

$$|\Delta x|=|x^*-x|=|x|\cdot\frac{|x^*-x|}{|x|}=|x|\cdot|e_x|\leqslant|x|\cdot\delta_x=$$

$$[(x_1\times 10^{-1}+x_2\times 10^{-2}+\cdots+x_n\times 10^{-n})\times 10^m]\times\frac{1}{2(x_1+1)}\times 10^{-n+1}<$$

$$\frac{(x_1+1)\times 10^{m-1}}{2(x_1+1)}\times 10^{-n+1}=\frac{1}{2}\times 10^{m-n}$$

可见,近似值 x 的绝对误差限为$10^{m-n}/2$,故 x 至少具有 n 位有效数字。

从以上两个定理可知,有效数字位数可以刻画近似值的准确程度,绝对误差限与近似值有效数字的末位有关,相对误差限与近似值的有效数字位数有关。

例 1.5 用 3.141 6 来表示 π 的近似值时,它的相对误差限是多少?

解　近似值 3.141 6 有五位有效数字，并且左边第一位非零数字是 3，根据定理(1.1)，其相对误差限为

$$\delta=\frac{1}{2\times 3}\times 10^{-5+1}=\frac{1}{6}\times 10^{-4}$$

例 1.6　已知近似值 x 有两位有效数字，试求其相对误差限。

解　本题给出了近似值的有效数字位数，但没有给出第一位非零数字，遇到这种情况，可以按第一位非零数字为最不利情况考虑，故取 $x_1=1$。根据定理(1.1)，得

$$\delta_x=\frac{1}{2x_1}\times 10^{-n+1}=\frac{1}{2\times 1}\times 10^{-2+1}=5\%$$

即近似值 x 的相对误差限为 5%。

例 1.7　已知近似值 x 的相对误差限为 0.3%，问 x 至少有几位有效数字？

解　设 x 有 n 位有效数字。由于 x 的第一位非零数字 x_1 没有给出，故必须考虑其取 $1\leqslant x_1\leqslant 9$ 的各种情况。依据

$$0.3\%=\frac{3}{1\,000}=\frac{1}{\frac{10}{3}\times 10^2}>\frac{1}{4\times 10^2}=\frac{1}{2\times(1+1)}\times 10^{-2}$$

$$0.3\%=\frac{3}{1\,000}=\frac{1}{\frac{10}{3}\times 10^2}<\frac{1}{3\times 10^2}<\frac{1}{2\times 1}\times 10^{-2}=\delta_x$$

令 $1-n=-2$，则 $n=3$，则 x 至少具有 3 位有效数字。

例 1.8　为了使 $\sqrt{20}$ 的近似值的相对误差不超过 0.1%，问至少要取多少位有效数字？

解　根据定理(1.1)

$$|e_x|\leqslant\frac{1}{2x_1}\times 10^{-n+1}$$

因为 $\sqrt{20}$ 的第一个非零数字是 4，故有

$$|e_x|\leqslant\frac{1}{2\times 4}\times 10^{-n+1}\leqslant 0.1\%=10^{-3}$$

n 的最小值只能为 4。即 $\sqrt{20}$ 的近似值只要超过 4 位有效数字，则其相对误差就不大于 0.1%。

1.4　误差危害的防止措施

由于受计算机数据存储位数的限制，数值计算的每一步都可能出现舍入误差，而一个科学问题的解决，往往要经过成千上万次运算，误差的传递和累积可能会给计算结果带来严重影响。要对计算的每一步都进行误差分析，显然既不现实也不必要。人们通过对误差传递规律的分析，总结出数值计算过程中避免误差危害发生的几个原则，遵守这些原则将有助于提高计算结果的可靠性和降低误差危害出现的可能性。

1. 使用数值稳定的算法

所谓算法，就是给定一些数据后，按照一定规定顺序进行计算的一个运算序列，它是一个近似计算过程。要选择某个算法，就必须保证其计算结果能达到要求的精度。一般而言，初始

值的误差和计算中的舍入误差总是存在的，而数值计算也是逐步进行的，前一步计算结果的误差必然影响后一步计算结果的准确性。把数值计算过程中舍入误差不增长的算法称为数值稳定的，否则是数值不稳定的。只有数值稳定的算法才可能得出可靠的计算结果，数值不稳定的算法没有实用价值。

例 1.9 计算积分

$$E_n=\int_0^1 x^n e^{x-1}dx \quad (n=0,1,2,\cdots,9)$$

解 利用分步积分，得

$$E_n=\int_0^1 x^n e^{x-1}dx=[x^n e^{x-1}]_0^1-n\int_0^1 x^{n-1}e^{x-1}dx=1-nE_{n-1}$$

即得递推公式

$$E_n=1-nE_{n-1} \quad (n=1,2,\cdots,9) \tag{1.20}$$

其中 $E_0=1-e^{-1}$。利用递推公式(1.20)，保留小数点后六位的计算结果见表 1.2。

表 1.2

E_n	准确值	计算结果	绝对误差
E_1	0.367 879	0.367 879	0.000 000 44
E_2	0.264 241	0.264 242	−0.000 001
E_3	0.207 277	0.207 274	0.000 003
E_4	0.170 893	0.170 904	−0.000 011
E_5	0.145 533	0.145 480	0.000 053
E_6	0.126 802	0.127 120	−0.000 318
E_7	0.112 383	0.110 160	0.002 223
E_8	0.100 932	0.118 720	−0.017 788
E_9	0.091 612	−0.068 480	0.160 092

积分值本应为正，为何出现了负值？就是因为所用的递推公式是数值不稳定的，初始值的误差在传递过程中增长很快。若 E_0 的舍入误差是 σ，即

$$E_0=E_0^*+\sigma$$

则由递推公式有

$$E_1=1-E_0=1-E_0^*-\sigma=E_1^*-\sigma$$
$$E_2=1-2E_1=1-2E_1^*+2\sigma=E_2^*+2!\sigma$$
$$E_3=1-3E_2=1-3E_2^*-6\sigma=E_3^*-3!\sigma$$
$$E_4=1-4E_3=1-4E_3^*+4\times 3!\sigma=E_4^*+4!\sigma$$
$$\cdots\cdots$$
$$E_n=E_n^*+(-1)^n n!\sigma$$

取 E_0 的舍入误差 $\sigma=4.412\times10^{-7}$，计算到 E_9 所产生的误差为

$$9!\sigma\approx 0.160\ 1$$

而 E_9 保留三位有效数字的精确值为 0.091 6。显然，误差完全淹没了准确值，使计算结果严重偏离准确值。

如果将递推公式改写为

$$E_{n-1}=\frac{1-E_n}{n} \tag{1.21}$$

考虑到 $n\to\infty$ 时，$E_n\to 0$，取 $E_{20}=0$ 作为初值进行计算，得

$E_{20}=0.0$，$E_{19}=0.050\,000\,0$，$E_{18}=0.050\,000\,0$，…，$E_{10}=0.083\,877\,1$，$E_9=0.091\,612\,3$
得到了正确的结果。

用递推公式(1.20)计算时，每运算一步，初值的误差就放大 n 倍，误差逐步增大，因此算法是数值不稳定的，不能得到可靠结果；用递推公式(1.21)计算时，每运算一步，初值的误差就缩小到 $1/n$，误差逐步减小，算法是数值稳定的，能得到可靠的结果。这个例题说明，对于同一问题，选用的算法不同，结果会完全不同。在实际计算中，必须采用数值稳定的算法，不能采用数值不稳定的算法。

2. 避免两个相近的数相减

两个相近的数相减会造成有效数字位数严重减少，相对误差增大，影响计算结果精度，应尽量避免。

例如 $x=7.000\,002$，$y=6.999\,998$，它们都有 7 位有效数字，但 $x-y=0.000\,004$ 只剩下一位有效数字，使结果的相对误差明显增大。

例 1.10　当 $x=1\,000$ 时，计算 $\sqrt{x+1}-\sqrt{x}$ 的值。

解　**方法 1**：直接计算

$$\sqrt{x+1}-\sqrt{x}=\sqrt{1\,001}-\sqrt{1\,000}\approx 31.64-31.62=0.02$$

结果只有一位有效数字，相对误差较大。

方法 2：改变计算公式

$$\sqrt{x+1}-\sqrt{x}=\frac{1}{\sqrt{x+1}+\sqrt{x}}\approx 0.015\,81$$

结果有四位有效数字，相对误差较小。

例 1.11　计算 $A=10^7(1-\cos 2^\circ)$。

解　**方法 1**：直接计算

$$A\approx 10^7(1-0.999\,4)=6\times 10^3$$

结果只有一位有效数字。

方法 2：改变计算公式

$$A=10^7(1-\cos 2^\circ)=10^7\times 2\,(\sin 1^\circ)^2\approx 2\times 10^7\times 0.017\,45^2\approx 6.09\times 10^3$$

结果有三位有效数字。

可见，改变计算公式，可以有效避免两个相近数相减引起的有效数字位数减少，从而得到较高精度的结果。

一般地，当 $f(x)\approx f(x_0)$ 时，可以用泰勒级数展开式计算差值，即

$$f(x)-f(x_0)=f'(x_0)(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots \tag{1.22}$$

取右端的有限项作为左端的近似。

如果无法改变算法，就需要增加有效数字位数进行计算，这样会延长计算时间和占用更多计算机内存。

3. 绝对值太小的数不能作除数

在数值计算中，若用很小的数作除数，可能会因商太大而导致计算机溢出停机，而且若作为分母的这个小数有很小误差，就会对计算结果产生严重影响。

根据误差传递规律，两个近似值 x_1, x_2 的比值 x_1/x_2 的误差估计式为

$$\Delta\left(\frac{x_1}{x_2}\right) \approx \frac{|x_1|}{|x_2|^2}|\Delta x_2| + \frac{1}{|x_2|}|\Delta x_1|$$

若 $|x_2| \ll |x_1|$，则 $|x_1|/|x_2| \gg 1$，x_2 的误差就会被严重放大，给计算结果带来较大影响。因此不宜把绝对值太小的数作除数。

例如 $2.718\,2/0.001 = 2\,718.2$。如若分母变为 $0.001\,1$ 时，即分母有 $0.000\,1$ 的变化，比值变为 $2.718\,2/0.001\,1 \approx 2\,471.1$，有巨大变化。

4. 防止大数“吃掉”小数

在数值计算中，若参与加减运算的两个数的数量级相差很大，由于计算机做加减运算时要“对阶”(对阶是指两数进行加减运算时，必须使小数点对齐后才进行运算。对浮点数来说，小数点对齐就是使两个数的阶码相等。使两个数阶码相等的过程就称为对阶。对阶的过程，实际上都是采用小阶向大阶看齐的办法)，加之计算机的存储位数有限，可能造成绝对值小的数被绝对值大的数“吃掉”而不能发挥作用，严重影响计算结果的准确性，所以应采取措施防止大数“吃掉”小数。

例 1.12 在五位十进制计算机上计算 $A = 51\,234 + \sum_{i=1}^{1\,000}\delta_i$，其中 $\delta_i = 0.9$。

解 **方法 1**：先把参与运算的数 51 234 写成规格化形式 $51\,234 = 0.512\,34 \times 10^5$。“对阶”时把两数写成绝对值小于 1 而阶码相同的数，$\delta_i = 0.000\,009 \times 10^5$，五位十进制计算机只能存储五位小数，多余位略掉。计算过程为

$$\begin{aligned}A &= 0.512\,34 \times 10^5 + 0.000\,009 \times 10^5 + \cdots + 0.000\,009 \times 10^5 \triangleq \\ &[(0.512\,34 \times 10^5 + 0.000\,00 \times 10^5) + \cdots] + 0.000\,00 \times 10^5 \triangleq \\ &0.512\,34 \times 10^5\end{aligned}$$

符号“$\triangleq$”表示计算机中的运算。显然，计算结果是不可靠的。这是由于对阶时，与大数51 234相比，小数 δ_i 被当作零对待，被大数 51 234“吃掉”造成的。

方法 2：计算时先把数量级相同的 1 000 个 δ_i 加起来，再与 51 234 相加，就不会出现小数被大数“吃掉”的现象。此时

$$\begin{aligned}A &= 0.512\,34 \times 10^5 + 0.000\,009 \times 10^5 + \cdots + 0.000\,009 \times 10^5 = \\ &0.512\,34 \times 10^5 + (0.000\,009 \times 10^5 + \cdots + 0.000\,009 \times 10^5) \triangleq \\ &0.512\,34 \times 10^5 + 0.009\,00 \times 10^5 \triangleq \\ &0.521\,34 \times 10^5\end{aligned}$$

例 1.13 求解二次方程

$$x^2 - (10^9 + 1)x + 10^9 = 0$$

的根。

解 用因数分解容易得到方程的根是

$$x_1 = 10^9, \quad x_2 = 1$$

但用求根公式

$$x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

在能将规格化数表示到小数点后八位的计算机上编程计算时，首先要对阶：

$$-b=10^9+1=0.100\ 000\ 00\times10^{10}+0.000\ 000\ 00[01]\times10^{10}$$

由于等号右端第二项最后两位数字“01”在机器上表现不出来，实际上是

$$-b\triangleq10^9$$

类似地还有

$$\sqrt{b^2-4ac}\triangleq10^9$$

从而方程根的计算结果是

$$x_1\triangleq\frac{10^9+10^9}{2}=10^9,\quad x_2\triangleq\frac{10^9-10^9}{2}=0$$

显然，x_2 严重失真，这是大数“吃掉”小数的结果。

如果把 x_2 的公式写成

$$x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}=\frac{2c}{-b+\sqrt{b^2-4ac}}=\frac{2\times10^9}{10^9+10^9}=1$$

就能得到正常结果。

因此，在进行数值计算时，要分析参与运算的各个数的数量级，编程时对运算次序做出恰当安排，防止大数“吃掉”小数，保障计算结果的可靠性。

5. 简化计算公式，减少运算次数

减少运算次数不仅能够减少运算量、提高运算速度，还可以减少误差传递次数，减少舍入误差的积累，提高结果精度。

例 1.14　分析计算多项式 $P_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n$ 的值时的运算次数。

解　若直接按上面公式逐项求积、求和计算，总共要作 $1+2+\cdots+n=n(n+1)/2$ 次乘法和 n 次加法。但如果将公式改写为下面的形式：

$$P_n(x)=a_0+(a_1+(a_2+\cdots+(a_{n-2}+(a_{n-1}+a_nx)x)x\cdots)x)x$$

即秦九韶算法，则只需 n 次乘法和 n 次加法就可以了。如 $n=20$，直接计算的运算次数是 230，秦九韶算法的运算次数是 40。

任何数值计算方法，只有在计算机上能快速得出可靠结果时，才是好的算法。通常从以下几个方面来评价一个数值计算方法的优劣：

(1) 计算量小；

(2) 计算结果误差小；

(3) 占用计算机内存少。

以上几个方面往往难以兼顾，实际计算时要根据精度要求，结合计算机性能特点，选择合适的数值计算方法，以便快速得出满意结果。

习　　题

1. 根据误差的来源，把误差主要分为哪四类？分别给出它们的定义，并举例说明。
2. 什么是有效数字？近似值的有效数字位数与其绝对误差限和相对误差限分别有何关

系？试举例说明。

3. 在数值计算中，防止误差危害发生的主要措施有哪五个？

4. 采用“四舍五入”规则，分别给出下列各准确值的具有四位有效数字的近似值，并指出近似值的绝对误差限。

$$1.202\,1,\quad 1.000\,4,\quad 76.450,\quad 0.001\,271\,3,\quad 0.001\,000\,8$$

5. 分别给出下列各近似值的有效数字位数，并计算其绝对误差限和相对误差限。

$$x_1=-1.002\,10,\quad x_2=0.032,\quad x_3=-0.040\,10$$

6. 下列各近似值均有四位有效数字：

$$x_1=0.012\,34,\quad x_2=123.4,\quad x_3=-1.010$$

试指出它们各自的绝对误差限和相对误差限。

7. 要使圆周率 π 的近似值的相对误差不超过 0.01%，问该近似值至少要有几位有效数字？

8. 已知 $\sqrt{3}=1.732\,050\,8\cdots$，试给出其具有 3,4,5 位有效数字的近似值，并求出相应的绝对误差限和相对误差限。

9. 数列 x_n 的递推公式是 $x_n=10x_{n-1}-1(n=1,2,\cdots)$。若取 $x_0=\sqrt{2}\approx1.41$（三位有效数字），问按这个递推公式求得的 x_{10} 的误差有多大？这个计算过程是数值稳定的吗？

10. 取 $\sqrt{2}\approx1.4$，分别应用下列各式计算 $f=(\sqrt{2}-1)^6$ 的值。

(1) $\dfrac{1}{(\sqrt{2}+1)^6}$；(2) $(3-2\sqrt{2})^3$；(3) $\dfrac{1}{(3+2\sqrt{2})^3}$；(4) $99-70\sqrt{2}$。

哪一个公式得到的结果最好？

11. 试改变以下各表达式，使其计算结果更为准确。

(1) $\dfrac{1}{1+2x}-\dfrac{1-x}{1+x}$, $|x|\ll1$；

(2) $\sqrt{x+\dfrac{1}{x}}-\sqrt{x-\dfrac{1}{x}}$, $|x|\gg1$；

(3) $\dfrac{1-\cos x}{x}$, $|x|\ll1$ 且 $x\neq0$。

12. 求方程 $x^2-56x+1=0$ 的两个根，使它们至少具有四位有效数字（已知 $\sqrt{3\,132}\approx55.964\,27$）。

13. 设近似值 $x>0$，并且其相对误差限是 δ，求 $\ln x$ 的相对误差限。

第 2 章　函数的插值与拟合及快速傅里叶变换

在实际问题中，变量 x 与 y 之间的函数关系 $y=f(x)$ 是事先不知道的，人们往往通过实验测量得出与变量 x 的离散值 $x_i(i=0,1,2,\cdots,n)$ 对应的变量 y 的离散值 $y_i(i=0,1,2,\cdots,n)$，并以数表的形式给出。这种表格形式的函数，既不利于探讨变量 x 与 y 之间的数学关系，也不便于计算表中没有给出的函数值。因此，有必要建立一些方法，能够根据已知的 x_i 与 $y_i(i=0,1,2,\cdots,n)$ 之间的对应关系，确定函数关系 $y=f(x)$ 的近似形式。函数的插值和拟合正是两种常用的解决这类问题的方法。函数的插值和拟合就是用既易于计算又能反映函数 $y=f(x)$ 主要特征的简单函数来近似函数 $y=f(x)$。由于采用的近似标准不同、所选用的简单函数的类型不同以及构造方法不同，便产生了不同类型的插值方法和拟合方法。插值和拟合还是物理实验数据的常用处理方法。

傅里叶变换是信号处理和数据处理的重要工具，在力学、波动学、声学、光学和通信工程中广泛应用。它使时域或空域问题的频域分析成为可能，可以简化研究过程，更加深刻地揭示问题的本质。随着计算机技术的进步，作为傅里叶变换有效计算方法的快速傅里叶变换得到了长足发展，应用领域不断扩大。

本章主要介绍拉格朗日插值法、牛顿插值法及函数拟合的最小二乘法，同时简要介绍几种分段低次插值方法及快速傅里叶变换的理论基础和实现方法。

2.1　函数的插值

2.1.1　插值的基本概念

定义(2.1)　设函数 $y=f(x)$ 在区间 $[a,b]$ 上有定义，它在区间内的 $n+1$ 个互异点 $x_i(i=0,1,2,\cdots,n)$ 处的函数值 $f(x_i)$ 已知，记作

$$y_i=f(x_i)\quad(i=0,1,2,\cdots,n)\tag{2.1}$$

设 Φ 是给定的某个函数类，在 Φ 上选择简单函数 $y=\varphi(x)$ 作为函数 $y=f(x)$ 的近似函数，并且满足条件

$$\varphi(x_i)=f(x_i)\quad(i=0,1,2,\cdots,n)\tag{2.2}$$

这样的函数近似方法就称为函数插值。称式(2.2)为插值条件，称近似函数 $y=\varphi(x)$ 为函数 $y=f(x)$ 的插值函数，称函数 $y=f(x)$ 为被插值函数，称互异点 $x_i(i=0,1,2,\cdots,n)$ 为插值节点(简称"节点")，称 $(x_i,f(x_i))(i=0,1,2,\cdots,n)$ 为插值点，区间 $[a,b]$ 被称为插值区间。

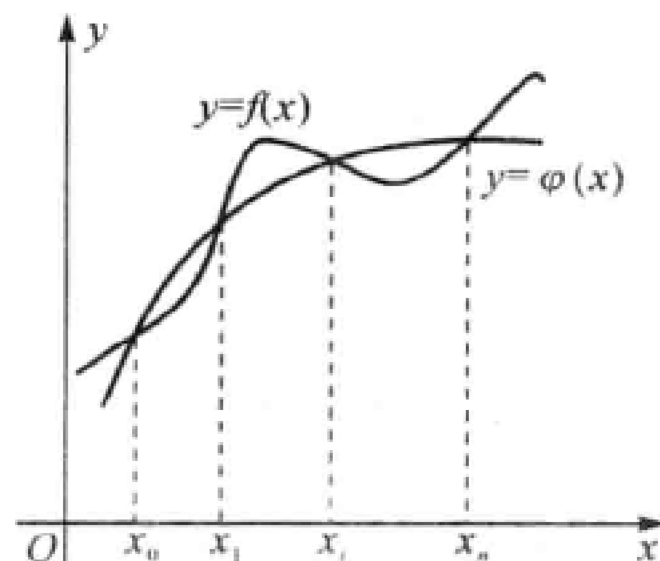


图　2.1

如图 2.1 所示，从几何上看，插值问题就是用经过点 $(x_i,f(x_i))(i=0,1,2,\cdots,n)$ 的曲线 $y=\varphi(x)$ 来近似同样经

过这些点的曲线 $y=f(x)$ 。

插值函数 $y=\varphi(x)$ 的选择要视具体问题而定，常用的插值函数类有代数多项式类、分段多项式类、有理函数类及三角函数类等。由于代数多项式具有简单和良好的性质，如无限光滑、容易计算导数和积分等，故常选用代数多项式作为插值函数，这样的插值方法称为多项式插值。

2.1.2　插值多项式的存在性与唯一性

定义(2.2)　一个在 $n+1$ 个互异节点 $x_i(i=0,1,2,\cdots,n)$ 上满足条件

$$P_n(x_i)=f(x_i)\quad(i=0,1,2,\cdots,n)\tag{2.3}$$

的不高于 n 次的多项式

$$P_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n\tag{2.4}$$

被称为插值多项式。

下面的定理(2.1)表明，插值多项式不仅是存在的，而且是唯一的。

定理(2.1)　在 $n+1$ 个互异节点 $x_i(i=0,1,2,\cdots,n)$ 处满足插值条件式(2.3)的次数不高于 n 次的插值多项式(2.4)存在且唯一。

证明　如果满足插值条件的插值多项式(2.4)的 $n+1$ 个系数 $a_i(i=0,1,2,\cdots,n)$ 可以被唯一确定，则该多项式也就存在并且唯一。

根据插值条件式(2.3)，插值多项式(2.4)的系数 $a_i(i=0,1,2,\cdots,n)$ 满足下面的 $n+1$ 阶线性方程组：

$$\left.\begin{aligned}a_0+a_1x_0+a_2x_0^2+\cdots+a_nx_0^n=y_0\\a_0+a_1x_1+a_2x_1^2+\cdots+a_nx_1^n=y_1\\\cdots\cdots\\a_0+a_1x_n+a_2x_n^2+\cdots+a_nx_n^n=y_n\end{aligned}\right\}\tag{2.5}$$

它是以 $a_i(i=0,1,2,\cdots,n)$ 为未知量的线性代数方程组，其系数行列式为范德蒙(Vandermonde)行列式

$$V=\begin{vmatrix}1&x_0&x_0^2&\cdots&x_0^n\\1&x_1&x_1^2&\cdots&x_1^n\\\vdots&\vdots&\vdots&&\vdots\\1&x_n&x_n^2&\cdots&x_n^n\end{vmatrix}=\prod_{n\geqslant i>j\geqslant 0}(x_i-x_j)\tag{2.6}$$

由于节点互异，即 $x_i\neq x_j(i\neq j)$，所以 $V\neq 0$。由克莱姆(Cramer)法则可知，方程组式(2.5)有唯一的一组解 $a_i(i=0,1,2,\cdots,n)$，也就是说插值多项式(2.4)被唯一确定，这就证明了 n 次代数多项式插值问题解的存在性与唯一性。

唯一性说明，无论用什么方法来构造插值多项式，也不管用什么形式来表示插值多项式，只要满足相同的插值条件，其结果都是相同的多项式。这给插值多项式的建立提供了方便。

2.1.3　插值多项式的余项

定义(2.3)　n 次插值多项式(2.4)给出的插值函数 $y=P_n(x)$ 只是函数 $y=f(x)$ 的近似，其误差被称为插值多项式的余项或截断误差，记作

$$R_n(x)=f(x)-P_n(x) \tag{2.7}$$

关于插值多项式的余项有以下定理，它是插值误差分析的基础。

定理(2.2)　设函数 $y=f(x)$ 在包含互异节点 $x_i(i=0,1,2,\cdots,n)$ 的最小开区间 I 上有直到 $n+1$ 阶的连续导数，则插值多项式(2.4) 的余项是

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x) \quad (\xi\in I) \tag{2.8}$$

式(2.8) 称为插值多项式的余项公式。其中

$$\omega_{n+1}(x)=\prod_{i=0}^{n}(x-x_i) \tag{2.9}$$

证明　当$x=x_i(i=0,1,2,\cdots,n)$ 时，定理显然成立。设 x 不同于所有节点 $x_i(i=0,1,2,\cdots,n)$，构造辅助函数

$$\varphi(t)=f(t)-P_n(t)-\frac{f(x)-P_n(x)}{\omega_{n+1}(x)}\omega_{n+1}(t)$$

显然，$\varphi(t)$ 有 $n+2$ 个互异零点 x 及 $x_i(i=0,1,2,\cdots,n)$。根据罗尔定理，$\varphi'(t)$ 在区间 I 内至少有 $n+1$ 个零点，$\varphi''(t)$ 在区间 I 内至少有 n 个零点，……。连续运用罗尔定理可知，至少存在一个点 $\xi\in I$，使 $\varphi^{(n+1)}(\xi)=0$。

注意到 $P_n(t)$ 是 n 次多项式，其 $n+1$ 阶导数等于零；$\omega_{n+1}(t)$ 是最高次项系数为 1 的 $n+1$ 次多项式，其 $n+1$ 阶导数等于 $(n+1)!$。于是

$$f^{(n+1)}(\xi)-\frac{f(x)-P_n(x)}{\omega_{n+1}(x)}(n+1)!\ =0$$

式中 $f(x)-P_n(x)=R_n(x)$，整理后得式(2.8)。

令 $M=\max\limits_{x\in I}|f^{(n+1)}(x)|$，则有

$$|R_n(x)|\leqslant\frac{M}{(n+1)!}\prod_{i=0}^{n}|x-x_i| \tag{2.10}$$

由于 ξ 无法具体求出，因此常用上式来估计插值多项式的截断误差。

如果被插值函数 $f(x)$ 本身就是不高于 n 次的多项式，由余项公式(2.8) 可知 $R_n(x)\equiv 0$，因此 $f(x)$ 的 n 次插值多项式 $P_n(x)$ 与 $f(x)$ 本身完全相等。

从理论上讲，从方程组式(2.5) 求出 $a_i(i=0,1,2,\cdots,n)$ 后就得到了插值多项式(2.4)。但由于解线性方程组的计算量往往很大，所以通常不使用这种方法建立插值多项式，而用构造法。下面分别介绍拉格朗日插值法和牛顿插值法。

2.2　拉格朗日插值法

2.2.1　拉格朗日插值多项式的构造

先建立多项式 $l_k(x)$，使它在节点 $x_i(i=0,1,2,\cdots,n)$ 处的值为

$$l_k(x_i)=\begin{cases}1 & (i=k)\\ 0 & (i\neq k)\end{cases}\quad (i,k=0,1,2,\cdots,n) \tag{2.11}$$

由条件 $l_k(x_i)=0(i\neq k)$ 可知，$x_i(i=0,1,2,\cdots,k-1,k+1,\cdots,n)$ 都是 $l_k(x)$ 的零点，因此能够假设

$$l_k(x)=A_k(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)$$

其中 A_k 是待定常数。再根据条件 $l_k(x_k)=1$ 和 $x_i(i=0,1,2,\cdots,n)$ 互异可求出 A_k，即

$$A_k=\frac{1}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$$

于是

$$l_k(x)=\frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}=\prod_{\substack{j=0\\j\neq k}}^{n}\frac{(x-x_j)}{(x_k-x_j)} \quad (2.12)$$

定义(2.4)　公式(2.12)表示的函数 $l_k(x)$ $(k=0,1,2,\cdots,n)$ 称为以 $x_i(i=0,1,\cdots,n)$ 为节点的n次基本插值多项式或n次拉格朗日插值基函数。

以 $n+1$ 个 n 次拉格朗日插值基函数 $l_k(x)$ $(k=0,1,2,\cdots,n)$ 为基础，就能写出满足插值条件式(2.3)的 n 次插值多项式。

定理(2.3)　n 次多项式插值问题的解可以表示为

$$P_n(x)=l_0(x)f(x_0)+l_1(x)f(x_1)+\cdots+l_n(x)f(x_n)=\sum_{i=0}^{n}l_i(x)f(x_i) \quad (2.13)$$

证明　因为 $l_k(x)$ $(k=0,1,2,\cdots,n)$ 是n次多项式，所以 $P_n(x)$ 是次数不超过n次的多项式。再根据

$$l_k(x_i)=\begin{cases}1 & (i=k)\\0 & (i\neq k)\end{cases} \quad (i,k=0,1,2,\cdots,n)$$

有

$$P_n(x_i)=\sum_{k=0}^{n}l_k(x_i)f(x_k)=f(x_i) \quad (i=0,1,2,\cdots,n)$$

结合插值多项式的唯一性，公式(2.13)就是满足插值条件式(2.3)的 n 次插值多项式。

定义(2.5)　称公式(2.13)表示的 n 次插值多项式为拉格朗日插值多项式。记为

$$L_n(x)=\sum_{i=0}^{n}l_i(x)f(x_i) \quad (2.14)$$

由插值多项式的唯一存在性可以断定，式(2.14)必与解方程组式(2.5)所确定的插值多项式(2.4)完全相同，只是表示形式不同。同样，拉格朗日插值多项式的余项或截断误差也如式(2.8)。以上构造插值多项式的方法称为基函数法。

利用式(2.9)，可以把 $L_n(x)$ 写成另外一种形式。对式(2.9)两边取对数，得

$$\ln\omega_{n+1}(x)=\sum_{i=0}^{n}\ln(x-x_i)$$

再对 x 求导数得

$$\frac{\omega'_{n+1}(x)}{\omega_{n+1}(x)}=\sum_{i=0}^{n}\frac{1}{(x-x_i)}$$

于是

$$\omega'_{n+1}(x)=\sum_{i=0}^{n}\frac{\omega_{n+1}(x)}{(x-x_i)}=\sum_{i=0}^{n}(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n) \quad (2.15)$$

$$\omega'_{n+1}(x_k)=(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n) \quad (2.16)$$

利用这些结果，公式(2.14)又可以改写为

$$L_n(x)=\sum_{i=0}^{n}\frac{\omega_{n+1}(x)}{(x-x_i)\omega'_{n+1}(x_i)}f(x_i) \tag{2.17}$$

令 $f(x)\equiv 1$，代入拉格朗日插值多项式(2.14)，可得出拉格朗日插值基函数的一个重要性质：

$$\sum_{k=0}^{n}l_k(x)\equiv 1 \tag{2.18}$$

从式(2.12)可以看出，拉格朗日插值基函数 $l_k(x)$ $(k=0,1,2,\cdots,n)$ 只与插值节点有关，与被插值函数 $f(x)$ 无关。从拉格朗日插值多项式(2.14)还能看出，拉格朗日插值多项式仅由插值点 $(x_i,f(x_i))$ $(i=0,1,2,\cdots,n)$ 确定，而与插值点的排列次序无关。

另外，为了便于编程计算，还常将 $L_n(x)$ 写成如下形式：

$$L_n(x)=\sum_{k=0}^{n}\left[\prod_{\substack{j=0\\j\neq k}}^{n}\frac{(x-x_j)}{(x_k-x_j)}\right]f(x_k) \tag{2.19}$$

编程时可用二重循环来完成对 $L_n(x)$ 的计算：先固定 k，令 j 从 0 到 n(但 $j\neq k$) 作乘积，然后再对 k 从 0 到 n 求和，即可得 $L_n(x)$ 的值。

2.2.2　线性插值多项式和二次插值多项式

1. 线性插值多项式

当 $n=1$ 时，插值点是 $(x_0,f(x_0))$ 和 $(x_1,f(x_1))$，拉格朗日插值多项式(2.14)为

$$L_1(x)=l_0(x)f(x_0)+l_1(x)f(x_1) \tag{2.20}$$

拉格朗日插值基函数式(2.12)为

$$l_0(x)=\frac{x-x_1}{x_0-x_1},\quad l_1(x)=\frac{x-x_0}{x_1-x_0} \tag{2.21}$$

从以上两式可得

$$L_1(x)=\frac{x-x_1}{x_0-x_1}f(x_0)+\frac{x-x_0}{x_1-x_0}f(x_1) \tag{2.22}$$

定义(2.6)　用式(2.22)表示的 $L_1(x)$ 作为 $f(x)$ 的近似函数时，称 $L_1(x)$ 为线性插值函数，也称 $L_1(x)$ 为线性插值多项式或一次插值多项式。

从几何上看，线性插值就是用过 $(x_0,f(x_0))$ 和 $(x_1,f(x_1))$ 两点的直线 $y=L_1(x)$ 来局部近似过这两点的曲线 $y=f(x)$，如图 2.2 所示。事实上，线性插值函数式(2.22)还能表示为

$$L_1(x)=f(x_0)+\frac{f(x_1)-f(x_0)}{x_1-x_0}(x-x_0)$$

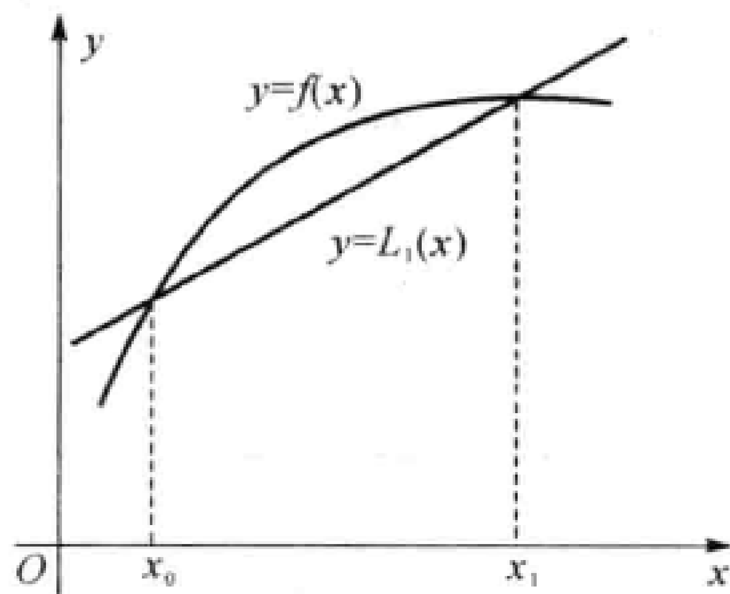


图　2.2

根据定理(2.2)，线性插值多项式(2.22)的余项是

$$R_1(x)=f(x)-L_1(x)=\frac{f''(\xi)}{2!}(x-x_0)(x-x_1)\quad(\xi\in[x_0,x_1]) \tag{2.23}$$

误差限表示为

$$|R_1(x)|\leqslant\frac{M}{2!}|(x-x_0)(x-x_1)|\quad(M=\max_{x\in[x_0,x_1]}|f''(x)|) \tag{2.24}$$

2. 二次插值多项式

当 $n=2$ 时，插值点是 $(x_0, f(x_0))$，$(x_1, f(x_1))$ 和 $(x_2, f(x_2))$，拉格朗日插值多项式 (2.14) 为

$$L_2(x)=l_0(x)f(x_0)+l_1(x)f(x_1)+l_2(x)f(x_2) \tag{2.25}$$

拉格朗日插值基函数式(2.12) 为

$$l_0(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)},\quad l_1(x)=\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)},\quad l_2(x)=\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \tag{2.26}$$

从以上两式可得

$$L_2(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0)+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1)+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2) \tag{2.27}$$

定义(2.7) 用式(2.27) 表示的 $L_2(x)$ 作为 $f(x)$ 的近似函数时，称 $L_2(x)$ 为二次插值或抛物线插值函数，也称 $L_2(x)$ 为二次插值多项式。

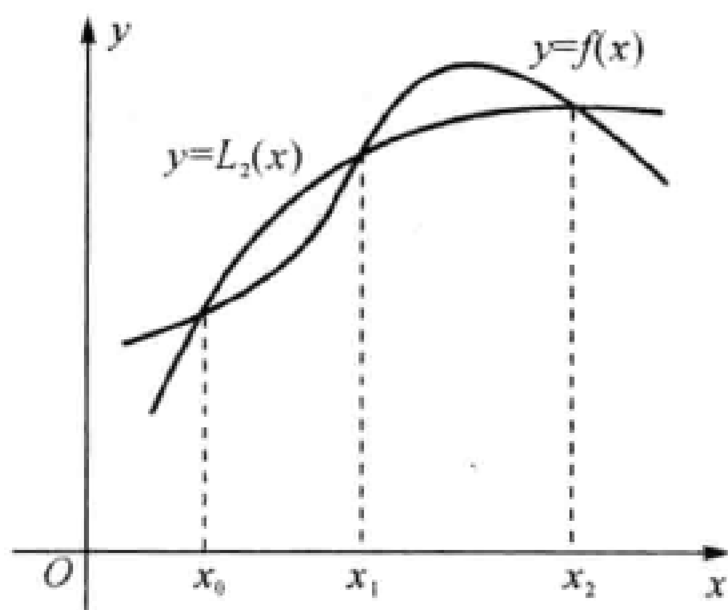


图 2.3

从几何上看，二次插值就是用过 $(x_0, f(x_0))$，$(x_1, f(x_1))$ 和 $(x_2, f(x_2))$ 三个点的抛物线 $y=L_2(x)$ 来局部近似过这三点的曲线 $y=f(x)$，如图 2.3 所示。

二次插值多项式的余项是

$$R_2(x)=f(x)-L_2(x)=\frac{f^{(3)}(\xi)}{3!}(x-x_0)(x-x_1)(x-x_2) \tag{2.28}$$

式中，$\xi\in\bar{I}$($\bar{I}$ 为包括 x_0, x_1, x_2 的最小闭区间)。误差限表示为

$$|R_2(x)|\leqslant\frac{M}{3!}|(x-x_0)(x-x_1)(x-x_2)|\quad (M=\max_{x\in\bar{I}}|f^{(3)}(x)|) \tag{2.29}$$

例 2.1 已知函数 $y=\ln x$ 的函数表，见表 2.1。试分别用线性插值和抛物线插值求 ln11.75 的近似值，并估计截断误差。

表 2.1

x	10	11	12	13	14
y	2.302 6	2.397 9	2.484 9	2.564 9	2.639 1

解 根据截断误差的特点，要减少截断误差，在选择插值节点时，就应该尽量选取与插值点 x 较近的那些节点。本题中 $x=11.75$ 介于节点 11 与 12 之间，故作线性插值时取节点 $x_0=11$，$x_1=12$。用线性插值公式(2.22) 有

$$L_1(x)=\frac{x-12}{11-12}\times 2.397\,9+\frac{x-11}{12-11}\times 2.484\,9=0.087x+1.440\,9$$

将 $x=11.75$ 代入，即得

$$\ln 11.75\approx L_1(11.75)\approx 2.463\,2$$

根据线性插值的余项公式(2.23)，有

$$R_1(x)=\frac{f''(\xi)}{2!}(x-x_0)(x-x_1)\quad(\xi\in[x_0,x_1])$$

本题中 $f(x)=\ln x$，而 $f''(x)=-\frac{1}{x^2}$，ξ 在 11 与 12 之间，所以

$$|f''(\xi)|=\frac{1}{\xi^2}\leqslant\frac{1}{11^2}$$

因此用线性插值计算 ln11.75 时的截断误差满足

$$|R_1(11.75)|\leqslant\frac{1}{2!\times 11^2}|(11.75-11)(11.75-12)|<0.000\ 8$$

实际误差是

$$\ln 11.75-2.463\ 2=0.000\ 7$$

在误差限 0.000 8 以内。

类似地，在用抛物线插值计算 ln11.75 时，选取节点 $x_0=11$，$x_1=12$，$x_2=13$，得到的插值多项式、近似值和截断误差分别是

$$L_2(x)=\frac{(x-12)(x-13)}{(11-12)(11-13)}\times 2.397\ 9+\frac{(x-11)(x-13)}{(12-11)(12-13)}\times 2.484\ 9+\frac{(x-11)(x-12)}{(13-11)(13-12)}\times 2.564\ 9=-0.003\ 5x^2+0.167\ 5x+0.978\ 9$$

$$\ln 11.75\approx L_2(11.75)\approx 2.463\ 8$$

$$|R_2(11.75)|\leqslant\frac{2}{3!\times 11^3}|(11.75-11)(11.75-12)(11.75-13)|<0.000\ 06$$

实际误差是

$$\ln 11.75-2.463\ 8=0.000\ 05$$

在误差限 0.000 06 以内。显然，抛物线插值的误差比线性插值的误差小很多。

例 2.2　要制作正弦函数 $\sin x$ 的函数表，已知的表中数值有四位小数，要求用线性插值计算的函数近似值的误差不超过表中数值的舍入误差，试求最大允许插值区间。

解　设最大允许插值区间为 $h=x_{i+1}-x_i$，依据线性插值误差公式(2.23)，有

$$|R_1(x)|=\left|\frac{f''(\xi)}{2!}(x-x_{i+1})(x-x_i)\right|=\left|\frac{\sin\xi}{2}(x-x_{i+1})(x-x_i)\right|\leqslant\left|\frac{1}{2}(x-x_{i+1})(x-x_i)\right|\leqslant\left|\frac{1}{2}\left(\frac{x_{i+1}+x_i}{2}-x_{i+1}\right)\left(\frac{x_{i+1}+x_i}{2}-x_i\right)\right|=\frac{1}{8}|(x_i-x_{i+1})(x_{i+1}-x_i)|=\frac{h^2}{8}\leqslant\frac{1}{2}\times 10^{-4}$$

解得 $h\leqslant 0.02$，即最大允许插值区间为 0.02。

程序(2.1)　建立拉格朗日插值多项式的 MATLAB 程序。

程序任务：建立基于 $(n+1)$ 个插值点 (x_i, y_i) 的 n 次拉格朗日插值多项式。

```
function [c, l] = lagrangepoly(x, y)
% 输入:x—— 插值节点 x_i 组成的行向量((n+1)×1)
%      y—— 插值节点处的函数值 y_i = f(x_i) 组成的行向量((n+1)×1)
% 输出:c—— 拉格朗日插值多项式的系数行向量[a_n  a_{n-1}  …  a_0]
%      l—— 拉格朗日插值基函数的系数矩阵(以降次排列)
w = length(x);             % 向量 x 的长度
```

```
n = w - 1;                  % 插值多项式的次数
l = zeros(w, w);            % 初始化矩阵 l
for k = 1: (n+1)            % 建立拉格朗日插值基函数
    v = 1;
    for j = 1: (n+1)
      if k ~= j
           v = conv(v, poly(x(j)))/(x(k) - x(j));   % 计算第 k 个基函数的系数向量
      end
    end
    l(k,:) = v;          % 存储第 k 个基函数的系数向量
end
c = y * l;               % 计算拉格朗日插值多项式的系数向量
```

在 MATLAB 命令窗口输入：

```
>> x=1:6;                              % 输入插值节点向量
>> y=[16, 18, 21, 17, 15, 12];         % 输入插值节点处的函数值向量
>> [c, l]=lagrangepoly(x, y);          % 调用程序(2.1)计算插值多项式系数向量 c
```

输出结果：

```
>> c = -0.2417   4.3333   -28.9583   87.6667   -115.8000   69.0000
```

对应的拉格朗日插值多项式为

$$L_5(x) = -0.2417x^5 + 4.3333x^4 - 28.9583x^3 + 87.6667x^2 - 115.8000x + 69.0000$$

继续在 MATLAB 命令窗口输入：

```
>> u=0.75:0.05:6.25;                   % 输入坐标点向量
>> v=polyval(c,u);                     % 计算插值函数在坐标点处的值向量
>> plot(x,y,':k',u,v,'-k')             % 作图
```

输出结果见图 2.4。图中折线是线性插值结果，曲线是 5 次多项式的插值结果，直线与曲线的交点是插值点。

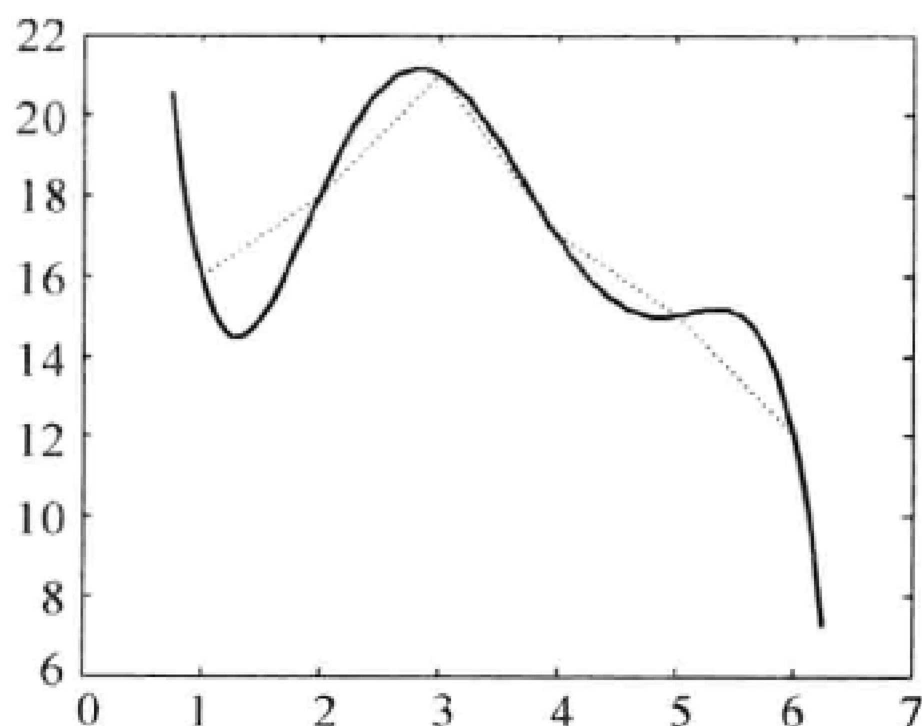


图 2.4

程序(2.2) 计算拉格朗日插值法插值的 MATLAB 程序。

程序任务：基于 n 个点 (x_i, y_i) 建立 $(n-1)$ 次拉格朗日插值多项式，计算插值多项式在坐

标点 xi 处的值。

```
function yi = Lagrangeinter(x, y, xi)
% 输入:x—— 插值节点向量(n×1)
%      y—— 插值节点处的函数值向量(n×1)
%      xi—— 坐标点向量
% 输出:yi—— 坐标点处的插值向量
m = length(x); n = length(y); p = length(xi);
if m ~= n error('向量 x 与 y 的长度必须一致');end
s = 0;
for k = 1:n
    t = ones(1,p);
    for j = 1: n
        if j ~= k
            t = t. * (xi - x(j))/(x(k) - x(j));% 计算 l_i(xi),即基函数在插值点的值
        end
    end
    s = s + t * y(k);                % 计算 L_n(xi) = Σ_{i=0}^{n} l_i(xi) f(x_i)
end
yi = s;
```

在 MATLAB 命令窗口输入：

```
>> x =[100,121];                      % 输入插值节点向量
>> y =[10,11];                        % 输入插值节点处的函数值向量
>> yi =Lagrangeinter(x, y, 115);      % 调用程序(2.2)计算 xi =115 处的插值函数值
>> yi                                 % 输出 yi 的计算结果
```

输出结果：

```
yi =10.7143
```

在 MATLAB 命令窗口输入：

```
>> x =[100,121];                      % 输入插值节点向量
>> y =[10,11];                        % 输入插值节点处的函数值向量
>> xi =[115,125];                     % 定义坐标点向量
>> yi =Lagrangeinter(x,y,xi);         % 调用程序(2.2)计算插值函数值向量
>> yi                                 % 输出计算结果
```

输出结果：

```
yi =10.7143    11.1905
```

2.3　牛顿插值法

拉格朗日插值公式直观、简单、容易构造，非常适合在计算机上使用。但拉格朗日插值公式的一个明显不足是，每个插值基函数都与所有节点有关，当增加一个新的插值节点时，除了

插值多项式要增加一项外，插值多项式的每一项也要重新建立，插值多项式必须全部重新构造。为了克服这一缺点，本节学习能够方便增加插值节点的牛顿插值公式。作为预备，首先介绍差商的概念。

2.3.1 差商

定义(2.8) 设函数 $f(x)$ 在 $n+1$ 个互异节点 $x_i(i=0,1,2,\cdots,n)$ 上的值分别为 $f(x_i)$，称

$$f[x_i]=f(x_i) \tag{2.30}$$

为函数 $f(x)$ 关于节点 x_i 的零阶差商；称

$$f[x_i,x_j]=\frac{f[x_i]-f[x_j]}{x_i-x_j} \tag{2.31}$$

为函数 $f(x)$ 关于节点 x_i 和 x_j 的一阶差商；称

$$f[x_i,x_j,x_k]=\frac{f[x_i,x_j]-f[x_j,x_k]}{x_i-x_k} \tag{2.32}$$

为函数 $f(x)$ 关于节点 x_i,x_j 和 x_k 的二阶差商。

一般地，把 $k-1$ 阶差商的差商

$$f[x_i,x_{i+1},\cdots,x_{i+k}]=\frac{f[x_i,x_{i+1},\cdots,x_{i+k-2},x_{i+k}]-f[x_i,x_{i+1},\cdots,x_{i+k-2},x_{i+k-1}]}{x_{i+k}-x_{i+k-1}} \tag{2.33}$$

称为函数 $f(x)$ 关于节点 $x_i,x_{i+1},\cdots,x_{i+k}$ 的k 阶差商。

定理(2.4) 差商具有以下基本性质：

(1) 差商与节点的排列次序无关。也就是说，任意调换节点次序，差商的值不变。即

$$f[x_0,\cdots,x_i,\cdots,x_j,\cdots,x_k]=f[x_0,\cdots,x_j,\cdots,x_i,\cdots,x_k]$$

(2) 差商可以表示为节点处函数值的线性组合。即

$$f[x_0,x_1,\cdots,x_k]=\sum_{i=0}^{k}\frac{f(x_i)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_k)}=\sum_{i=0}^{k}\frac{f(x_i)}{\omega'_{k+1}(x_i)} \tag{2.34}$$

(3) 如果函数 $f(x)$ 的 k 阶差商 $f[x_0,x_1,\cdots,x_{k-1},x]$ 是 x 的 m 次多项式，则其 $k+1$ 阶差商 $f[x_0,x_1,\cdots,x_k,x]$ 是 x 的 $m-1$ 次多项式。

事实上，根据差商的定义

$$f[x_0,x_1,\cdots,x_{k-1},x_k,x]=\frac{f[x_0,x_1,\cdots,x_{k-1},x]-f[x_0,x_1,\cdots,x_{k-1},x_k]}{x-x_k}$$

上式的分子是 x 的 m 次多项式，且当 $x=x_k$ 时其值为零，即分子中含有因子 $(x-x_k)$，与分母中的相同因子约去后，上式就成为 x 的 $m-1$ 次多项式。

由此可见，当 $f(x)$ 是 n 次多项式时，其 k 阶差商 $f[x_0,x_1,\cdots,x_{k-1},x]$ 在 $k\leqslant n$ 时是 $n-k$ 次多项式，在 $k=n$ 时等于常数，在 $k>n$ 时等于零。

(4) 设 $f(x)$ 在包含互异节点 $x_0,x_1,\cdots,x_n$ 的闭区间 $[a,b]$ 上有 n 阶导数，则在这一区间内至少有一点 ξ，满足

$$f[x_0,x_1,\cdots,x_n]=\frac{f^{(n)}(\xi)}{n!}\quad \xi\in(a,b) \tag{2.35}$$

上式反映了差商与导数之间的关系。

2.3.2 牛顿插值公式

设 $x \in [a,b]$，由一阶差商公式

$$f[x,x_0]=\frac{f(x)-f(x_0)}{x-x_0}$$

可得

$$f(x)=f(x_0)+f[x,x_0](x-x_0) \tag{2.36}$$

由二阶差商公式

$$f[x,x_0,x_1]=\frac{f[x,x_0]-f[x_0,x_1]}{x-x_1}$$

可得

$$f[x,x_0]=f[x_0,x_1]+f[x,x_0,x_1](x-x_1)$$

将其代入式(2.36)，有

$$f(x)=f(x_0)+f[x_0,x_1](x-x_0)+f[x,x_0,x_1](x-x_0)(x-x_1) \tag{2.37}$$

重复以上过程，能够推导出

$$\begin{aligned} f(x)=&f(x_0)+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)+\cdots+\\ &f[x_0,x_1,\cdots,x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1})+\\ &f[x,x_0,x_1,\cdots,x_n](x-x_0)(x-x_1)\cdots(x-x_n) \end{aligned} \tag{2.38}$$

定义(2.9)　设函数 $f(x)$ 在 $n+1$ 个互异节点 $x_i(i=0,1,2,\cdots,n)$ 上的值分别为 $f(x_i)$，函数

$$\begin{aligned} N_n(x)=&f(x_0)+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)+\\ &f[x_0,x_1,\cdots,x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1}) \end{aligned} \tag{2.39}$$

和

$$\begin{aligned} R_n(x)=&f[x,x_0,x_1,\cdots,x_n](x-x_0)(x-x_1)\cdots(x-x_n)=f[x,x_0,x_1,\cdots,x_n]\omega_{n+1}(x)=\\ &\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x-x_i)\quad(\xi\in I) \end{aligned} \tag{2.40}$$

分别称为函数 $f(x)$ 的n 次牛顿插值多项式和余项公式。

显然，函数 $f(x)$ 可以表示为

$$f(x)=N_n(x)+R_n(x)$$

余项公式(2.40) 满足

$$R_n(x_i)=0\quad(i=0,1,2,\cdots,n)$$

牛顿插值多项式(2.39) 满足插值条件

$$N_n(x_i)=f(x_i)\quad(i=0,1,2,\cdots,n)$$

且次数不超过 n 次。根据插值多项式的唯一性，必有 $N_n(x)\equiv L_n(x)\equiv P_n(x)$，$N_n(x)$ 只是形式上与拉格朗日插值多项式 $L_n(x)$ 不同。

公式(2.39) 表明，牛顿插值多项式各项的系数就是函数 $f(x)$ 的各阶差商

$$f[x_0],f[x_0,x_1],f[x_0,x_1,x_2],\cdots,f[x_0,x_1,\cdots,x_n]$$

因此，在构造牛顿插值公式时，通常先把差商列成表格的形式(见表 2.2)，称为差商表。

表 2.2

i	x_i	零阶差商	一阶差商	二阶差商	三阶差商
0	x_0	$f[x_0]$			
			$f[x_0,x_1]$		
1	x_1	$f[x_1]$		$f[x_0,x_1,x_2]$	
			$f[x_1,x_2]$		$f[x_0,x_1,x_2,x_3]$
2	x_2	$f[x_2]$		$f[x_1,x_2,x_3]$	
			$f[x_2,x_3]$		$f[x_1,x_2,x_3,x_4]$
3	x_3	$f[x_3]$		$f[x_2,x_3,x_4]$	
			$f[x_3,x_4]$		
⋮	⋮	⋮	⋮	⋮	⋮

另外，由差商的特点可知，如果一个函数所有 k 阶差商的值相等，那么用 k 次多项式近似它是合适的。

例 2.3 已知函数 $f(x)=\text{sh}x$ 的函数表，见表 2.3。构造四次牛顿插值多项式，并计算 $f(0.596)$ 的值。

表 2.3

i	0	1	2	3	4	5
x_i	0.40	0.55	0.65	0.80	0.90	1.05
$f(x_i)$	0.410 75	0.578 15	0.696 75	0.888 11	1.026 52	1.253 86

解 由已知函数表构建的差商表见表 2.4。

表 2.4

x_i	$f(x_i)$	一阶差商	二阶差商	三阶差商	四阶差商
0.40	0.410 75				
		1.116 0			
0.55	0.578 15		0.280 0		
		1.186 0		0.197	
0.65	0.696 75		0.358 8		0.034
		1.275 7		0.214	
0.80	0.888 11		0.433 6		0.034
		1.384 1		0.231	
0.90	1.026 52		0.526 0		
		1.515 6			
1.05	1.253 86				

可见，四阶差商是常数，因此可以取四次插值多项式，其余项为零。选取 5 个插值节点 0.40～0.90，利用差商表中带阴影背景的数据，得四次牛顿插值多项式为

$$
\begin{aligned}
N_4(x)=&0.410\,75+1.116\,0(x-0.40)+0.280\,0(x-0.40)(x-0.55)+\\
&0.197(x-0.40)(x-0.55)(x-0.65)+\\
&0.034(x-0.40)(x-0.55)(x-0.65)(x-0.80)
\end{aligned}
$$

于是

$$f(0.596)=\mathrm{sh}(0.596)=N_4(0.596)=0.631\,92$$

拉格朗日插值法的优点是构造容易、形式对称、便于记忆。它的缺点是，如果增加插值节点，公式必须整个重写，这就增加了工作量。与拉格朗日插值法相比，牛顿插值法的优点是具有继承性，即增加新节点后，只要在原插值多项式中增加一项即可，不必全部重新构建。

2.3.3　差分与等距节点插值公式

等距节点的插值问题是一般插值问题的特殊情况，拉格朗日插值公式和牛顿插值公式对等距节点插值问题同样适用。但由于等距节点插值问题中相邻两个节点距离相同，引入差分概念后，等距节点插值公式能得到进一步简化。

定义(2.10)　已知函数 $f(x)$ 在等距节点 $x_i=x_0+hi\,(i=0,1,2,\cdots,n)$ 上的值 $f(x_i)=f_i$，这里常量 $h=x_{i+1}-x_i$ 称为步长。则称

$$\Delta f_i=f(x_i+h)-f(x_i)=f_{i+1}-f_i \tag{2.41}$$

为函数 $f(x)$ 在点 x_i 处步长为 h 的一阶向前差分。称

$$\Delta^2 f_i=\Delta(\Delta f_i)=\Delta f_{i+1}-\Delta f_i=f_{i+2}-2f_{i+1}+f_i \tag{2.42}$$

为函数 $f(x)$ 在点 x_i 处步长为 h 的二阶向前差分。

一般地，设 $k-1$ 阶差分已定义，则称

$$\Delta^k f_i=\Delta(\Delta^{k-1} f_i)=\Delta^{k-1} f_{i+1}-\Delta^{k-1} f_i=\sum_{j=0}^{k}(-1)^j\begin{pmatrix}k\\j\end{pmatrix}f_{i-j+k} \tag{2.43}$$

为函数 $f(x)$ 在点 x_i 处步长为 h 的 $k\,(k=2,3,\cdots)$ 阶向前差分。式中，$\begin{pmatrix}k\\j\end{pmatrix}=\dfrac{k!}{j!(k-j)!}$ 是二项式展开系数。规定

$$\Delta^0 f_i=f_i \tag{2.44}$$

为函数 $f(x)$ 在点 x_i 处的零阶差分。

类似地，也可以定义向后差分和中心差分。

定理(2.5)　向前差分有以下性质：

(1) 差分与差商的关系是

$$f[x_i,x_{i+1},\cdots,x_{i+j}]=\frac{1}{j!h^j}\Delta^j f_i \tag{2.45}$$

(2) 差分与导数的关系是，若函数 $f(x)$ 在包含等距节点 $x_i,x_{i+1},\cdots,x_{i+j}$ 的区间 I 上有 j 阶导数，则在这一区间内至少有一点 ξ，使得

$$\Delta^j f_i=h^j f^{(j)}(\xi) \tag{2.46}$$

可以用类似于编制差商表的方法构成差分表(见表 2.5)。

表 2.5

i	x_i	$\Delta^0 f_i$	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
0	x_0	f_0			
			$\Delta f_0 = f_1 - f_0$		
1	x_1	f_1		$\Delta^2 f_0 = \Delta f_1 - \Delta f_0$	
			$\Delta f_1 = f_2 - f_1$		$\Delta^3 f_0 = \Delta^2 f_1 - \Delta^2 f_0$
2	x_2	f_2		$\Delta^2 f_1 = \Delta f_2 - \Delta f_1$	
			$\Delta f_2 = f_3 - f_2$		
3	x_3	f_3			

当要计算点 x 处函数 $f(x)$ 的近似值而 x 在 x_0 与 x_1 之间时，可令 $x=x_0+th$，其中$0<t<1$。这样，公式(2.9)简化为

$$\omega_{i+1}(x)=t(t-1)(t-2)\cdots(t-i)h^{i+1} \tag{2.47}$$

将公式(2.45)和公式(2.47)代入牛顿插值公式(2.39)，则有

$$N_n(x)=f_0+t\Delta f_0+\frac{t(t-1)}{2!}\Delta^2 f_0+\cdots+\frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n f_0=$$
$$\sum_{j=0}^{n}\frac{t(t-1)\cdots(t-j+1)}{j!}\Delta^j f_0 \tag{2.48}$$

称为牛顿前插公式。当要计算的点 x 处在 x_{n-1} 与 x_n 之间时，可令 $x=x_n-th$，其中 $0<t<1$。同样能够推导出牛顿插值公式

$$N_n(x)=f_n-t\Delta f_{n-1}+\frac{t(t-1)}{2!}\Delta^2 f_{n-2}+\cdots+(-1)^n\frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n f_0=$$
$$\sum_{j=0}^{n}(-1)^j\frac{t(t-1)\cdots(t-j+1)}{j!}\Delta^j f_{n-j} \tag{2.49}$$

公式(2.48)和公式(2.49)都只用到向前差分，只是公式(2.48)用到向前差分表的前部，而公式(2.49)用到向前差分表的后部，所以分别被称为表前公式和表后公式。

插值公式(2.48)和公式(2.49)的余项均是

$$R_n(x)=h^{n+1}\frac{f^{(n+1)}(\xi)}{(n+1)!}t(t-1)(t-2)\cdots(t-n)\quad(\xi\in I) \tag{2.50}$$

例 2.4 已知等距节点及相应的函数值，见表 2.6。计算 $N_3(0.5)$ 和 $N_3(0.9)$ 的值。

表 2.6

x_i	0.4	0.6	0.8	1.0
$f(x_i)$	1.5	1.8	2.2	2.8

解 显然步长 $h=0.2$，该问题的向前差分表见表 2.7。

表　2.7

i	x_i	$\Delta^0 f_i$	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
0	0.4	1.5			
			0.3		
1	0.6	1.8		0.1	
			0.4		0.1
2	0.8	2.2		0.2	
			0.6		
3	1.0	2.8			

当 $x=0.5$ 时，$t=(x-x_0)/h=0.5$。将差分表中加阴影的差分值代入公式(2.48)，有

$$N_3(x)=1.5+0.3t+0.1\times\frac{t(t-1)}{2!}+0.1\times\frac{t(t-1)(t-2)}{3!}$$

代入 $t=0.5$，得到

$$N_3(0.5)=1.643\ 75$$

当 $x=0.9$ 时，$t=(x_3-x)/h=0.5$。将差分表中加下画线的差分值代入公式(2.49)，有

$$N_3(x)=2.8-0.6t+0.2\times\frac{t(t-1)}{2!}-0.1\times\frac{t(t-1)(t-2)}{3!}$$

代入 $t=0.5$，得到

$$N_3(0.9)=2.468\ 75$$

程序(2.3)　建立牛顿插值多项式的 MATLAB 程序。

程序任务：建立经过点 (x_k, y_k) $(k=0,1,\cdots,n)$ 的次数小于 n 的牛顿插值多项式

$$N_n(x)=d_{0,0}+d_{1,1}(x-x_0)+d_{2,2}(x-x_0)(x-x_1)+\cdots+$$
$$d_{n,n}(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

其中，$d_{k,0}=y_k$，$d_{k,j}=(d_{k,j-1}-d_{k-1,j-1})/(x_k-x_{k-j})$。

```
function [c, d] = newtonpoly(x, y)
% 输入：x—— 插值节点 x_i 组成的行向量((n+1)×1)
%       y—— 插值节点处函数值 y_i = f(x_i) 组成的行向量((n+1)×1)
% 输出：c—— 建立的牛顿插值多项式的系数行向量((n+1)×1)
%       d—— 建立的差商表
n = length(x);
d = zeros(n,n);
d(:,1) = y';      % 给差商表的第一列赋值
for j = 2:n       % 建立差商表
    for k = j:n
        d(k,j) = (d(k,j-1) - d(k-1,j-1))/(x(k) - x(k-j+1));
    end
end
c = d(n,n);       % 确定牛顿插值多项式的系数
```

```
for k = (n-1):-1:1
    c = conv(c, poly(x(k)));              % c*(x-x(k))
    m = length(c);
    c(m) = c(m) + d(k,k);
end
```

在 MATLAB 命令窗口输入：

```
>> x=1:6;                       % 输入插值节点向量
>> y=x.^3-4.*x;                 % 根据函数 y=x^3-4x 计算插值点的函数值向量
>> [c,d]=newtonpoly(x,y)        % 调用程序(2.3) 建立牛顿插值多项式和插商表
```

输出结果：

```
c=0  0  1  0  -4  0
```

对应的牛顿插值多项式为

$$N_5(x)=x^3-4x$$

建立的差商表是

```
d =
     -3    0    0   0   0   0
      0    3    0   0   0   0
     15   15    6   0   0   0
     48   33    9   1   0   0
    105   57   12   1   0   0
    192   87   15   1   0   0
```

程序(2.4) 等距节点牛顿向前插值的 MATLAB 程序。

程序任务：建立基于等距节点 $x_k(k=1,2,\cdots,n)$ 的向前牛顿插值多项式，并计算插值点 xi 处的插值。

```
function yi = newtonintf(x, y, xi)
% 输入：x—— 等距节点向量(n×1)
%       y—— 节点上的函数值向量(n×1)
%       xi—— 插值点(标量)
% 输出：yi—— 返回的插值
h = x(2) - x(1); t = (xi - x(1))/h;
% 建立差分表 Y
n = length(y); Y = zeros(n); Y(:, 1) = y';
for k = 1:n-1
    Y(:, k+1) = [diff(y', k); zeros(k, 1)];
end
% 计算向前插值结果
yi = Y(1,1);
for i = 1:n-1
    z = t;
    for k = 1:i-1
```

```
        z = z * (t - k);          %z = (t - 1)(t - 2)…(t - k)
    end
    yi = yi + Y(1, i + 1) * z/prod([1:i]); % 计算插值点的函数值
end
```

在 MATLAB 命令窗口输入：

```
>> x = 1:6;                                                    % 输入等距插值节点
>> y = [1.0000, 1.2599, 1.4422, 1.5874, 1.7100, 1.8171]; % 输入节点处的函数值
>> yi = newtonintf(x, y, 1.5);      % 调用程序(2.4)求插值点 xi = 1.5 处的插值
>> yi                                                          % 输出插值 yi
```

输出结果：

```
yi = 1.1437
```

程序(2.5)　等距节点牛顿向后插值的 MATLAB 程序。

程序任务：建立基于等距节点 $x_k(k=1,2,\cdots,n)$ 的向后的牛顿插值多项式，并计算插值点 xi 处的插值。

```
function yi = newtonintb(x, y, xi)
% 输入：x—— 等距节点向量(n × 1)
         y—— 节点上的函数值向量(n × 1)
         xi—— 插值点(标量)
% 输出：yi— 返回的插值
n = length(x); h = x(n) - x(n - 1); t = (x(n) - xi)/h;
n = length(y); Y = zeros(n); Y(:,1) = y';
for k = 1:n - 1          % 计算差分表 Y
    Y(:,k + 1) = [zeros(k,1); diff(y',k)];
end
yi = Y(n,1);             % 计算向后插值结果
for i = 1:n - 1
    z = t;
    for k = 1:i - 1
        z = z * (t - k);
    end
    yi = yi + Y(n,i + 1) * (- 1)^i * z/prod([1:i]);
end
```

在 MATLAB 命令窗口输入：

```
>> x = 1:6;                                                  % 输入等距插值节点
>> y = [1.0000,1.2599,1.4422,1.5874,1.7100,1.8171];         % 输入节点处的函数值
>> yi = newtonintb(x,y,5.6);         % 调用程序(2.5)求插值点 xi = 5.6 处的插值
>> yi                                                        % 输出插值 yi
```

输出结果：

```
yi = 1.7754
```

2.4 分段低次插值

人们通常有一种错觉，以为插值多项式的次数越高插值精度就越高，但事实并不是这样的。这是因为随着插值节点逐渐增加，插值多项式次数不断增高，插值多项式不一定收敛到被插值函数。德国数学家龙格（Runge）给出的一个典型例子是，对函数 $f(x)=1/(1+x^2)$，在区间$[-5,5]$上关于$(n+1)$个等距节点 $x_i=-5+10\times i/n(i=0,1,2,\cdots,n)$，作 n 次多项式插值 $P_n(x)$。随着n的增加，插值多项式 $P_n(x)$ 与函数$f(x)$的差异越来越严重，$n=10$时差异在 $x=\pm5$ 附近表现得十分明显，这种现象被称为龙格现象，如图 2.5 所示。

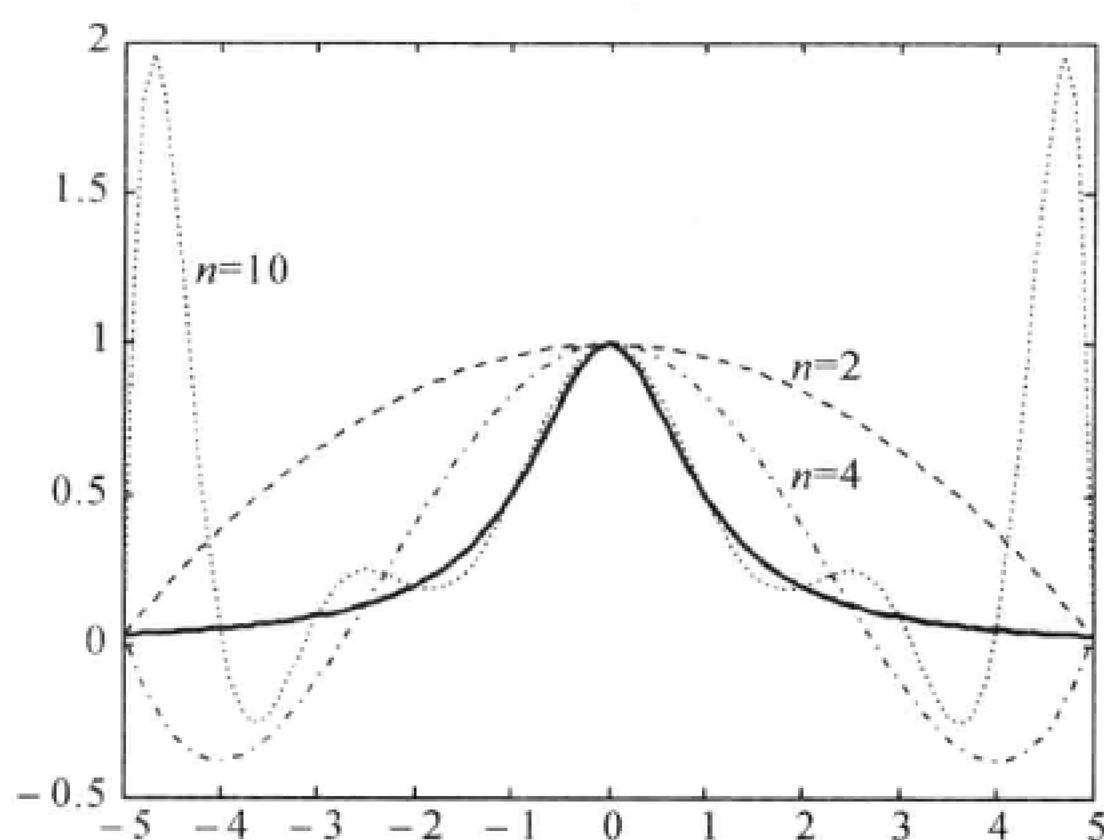


图 2.5

余项公式(2.8)表明，影响插值误差的因素有两个，一个是函数高阶导数$|f^{(n+1)}(\xi)|$，另一个是$|\omega_{n+1}(x)|$，它们任何一个增大，都可能导致插值误差的增大。许多函数导数的绝对值随着导数阶次的增加快速增加，因此随着插值多项式次数的升高，插值误差有可能增大。如函数 $f(x)=1/x$ 的导数 $f^{(n)}(x)=(-1)^n n!/x^{n+1}$，其绝对值按 $n!$速度增长。另外，对于固定的插值点 x，只有在离它较近的几个节点处，$|x-x_i|$的值才较小。随着节点的增多和节点离插值点距离的增大，$|x-x_i|$也会增大，可能导致$|\omega_{n+1}(x)|$过大，所以插值时选取的节点不宜离插值点太远。考虑到这两方面影响，在实际应用中较多用到一、二、三次多项式插值，高于七次的多项式插值几乎不用。

对于较大区间上的多节点插值问题，通常采用分段低次插值方法来处理。所谓分段低次插值就是用多个小区间上的低次插值函数来代替在整个区间上的高次插值函数。具体来说，就是先把较大插值区间分成若干小区间，在每个小区间上用低次插值函数（如线性插值多项式或二次插值多项式）进行插值；然后将每个小区间上的低次插值函数拼接起来，构成整个插值区间上的插值函数。

显然，分段低次插值具有公式简单、运算量小、稳定性好和精度要求有保证等优点。从理论上讲，只要每个小区间取得足够小，分段低次插值的误差总能控制在允许范围内。分段低次插值的明显缺点是难以保证整个插值曲线的光滑性。在相邻两段插值曲线的接点处两个插值函数的一阶导数不同，这对于有些实际问题是不能满足需要的。为了保证在整个插值区间上

插值函数低阶导数的连续性，就有了埃尔米特(Hermite)插值和样条函数插值等对函数导数有要求的插值方法。

2.4.1　分段线性插值和分段二次插值

从几何意义上说，分段线性插值就是用折线近似代替曲线 $y=f(x)$。先把区间 $[a,b]$ 分成以相邻两个节点为端点的小区间 $[x_i,x_{i+1}]$ $(i=0,1,\cdots,n-1)$，在区间 $[x_i,x_{i+1}]$ 上进行线性插值，插值函数为

$$I_i(x)=\frac{x-x_{i+1}}{x_i-x_{i+1}}f(x_i)+\frac{x-x_i}{x_{i+1}-x_i}f(x_{i+1})\quad (x\in[x_i,x_{i+1}]) \tag{2.51}$$

在整个区间 $[a,b]$ 上的插值函数为

$$I(x)=\begin{cases} I_0(x) & (x\in[x_0,x_1]) \\ I_1(x) & (x\in[x_1,x_2]) \\ \vdots & \vdots \\ I_{n-1}(x) & (x\in[x_{n-1},x_n]) \end{cases} \tag{2.52}$$

分段二次插值就是用分段抛物线近似代替曲线 $y=f(x)$。先把区间 $[a,b]$ 分成若干个小区间 $[x_{i-1},x_{i+1}]$，在小区间 $[x_{i-1},x_{i+1}]$ 上进行二次插值，插值函数为

$$\begin{aligned} Q_i(x)= & \frac{(x-x_i)(x-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})}f(x_{i-1})+\frac{(x-x_{i-1})(x-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})}f(x_i)+ \\ & \frac{(x-x_{i-1})(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}f(x_{i+1})\quad (x\in[x_{i-1},x_{i+1}]) \end{aligned} \tag{2.53}$$

在整个区间 $[a,b]$ 上的插值函数为

$$Q(x)=\begin{cases} Q_1(x) & (x\in[x_0,x_2]) \\ Q_3(x) & (x\in[x_2,x_4]) \\ \vdots & \vdots \\ Q_{n-1}(x) & (x\in[x_{n-2},x_n]) \end{cases} \tag{2.54}$$

2.4.2　分段三次埃尔米特多项式插值

埃尔米特插值不仅要求插值函数与被插值函数在节点处的值相等，而且还要求插值函数的一阶导数与被插值函数的一阶导数也相等。

定义(2.11)　设已知函数 $f(x)$ 在区间 $[a,b]$ 上 $n+1$ 个互异节点 $x_i(i=0,1,2,\cdots,n)$ 上的值 $f(x_i)$ 及一阶导数的值 $f'(x_i)$，要求构造一个插值函数 $H(x)$，使其满足条件：

(1) $H(x)$ 是一个次数不超过 $2n+1$ 次的多项式；

(2) $H(x_i)=f(x_i)$，　$H'(x_i)=f'(x_i)$　$(i=0,1,2,\cdots,n)$。

称此问题为埃尔米特插值问题，满足上述条件的插值多项式 $H(x)$ 称为埃尔米特插值多项式。

这里以分段三次埃尔米特多项式插值为例介绍埃尔米特插值的基本思路。分段三次埃尔米特插值就是把区间 $[a,b]$ 分成 n 个小区间 $[x_{i-1},x_i]$ $(i=1,2,\cdots,n)$，在每个区间 $[x_{i-1},x_i]$ 上用三次埃尔米特插值多项式 $H_3(x)$ 来近似被插值函数 $y=f(x)$。

在区间 $[x_{i-1},x_i]$ 上 $H_3(x)$ 满足

$$H_3(x_j)=f(x_j),\quad H'_3(x_j)=f'(x_j)\quad (j=i-1,i) \tag{2.55}$$

引入分段三次埃尔米特插值基函数 $\varphi_{i-1}(x)$，$\varphi_i(x)$，$\psi_{i-1}(x)$ 和 $\psi_i(x)$，在小区间 $[x_{i-1},x_i]$ 上，三次埃尔米特插值多项式 $H_3(x)$ 表示为

$$H_3(x)=\varphi_{i-1}(x)f(x_{i-1})+\varphi_i(x)f(x_i)+\psi_{i-1}(x)f'(x_{i-1})+\psi_i(x)f'(x_i) \tag{2.56}$$

能使 $H_3(x)$ 满足插值条件式(2.55)的分段三次埃尔米特插值基函数 $\varphi_{i-1}(x)$，$\varphi_i(x)$，$\psi_{i-1}(x)$ 和 $\psi_i(x)$ 分别是

$$\left.\begin{aligned}
\varphi_{i-1}(x)&=\frac{1}{h_i^3}(h_i-2x_{i-1}+2x)(x_i-x)^2\\
\varphi_i(x)&=\frac{1}{h_i^3}(h_i+2x_i-2x)(x-x_{i-1})^2\\
\psi_{i-1}(x)&=\frac{1}{h_i^2}(x-x_{i-1})(x-x_i)^2\\
\psi_i(x)&=\frac{1}{h_i^2}(x-x_i)(x-x_{i-1})^2
\end{aligned}\right\} \tag{2.57}$$

其中，区间长度 $h_i=x_i-x_{i-1}$。把上式给出的插值基函数代入式(2.56)，就得到小区间 $[x_{i-1},x_i]$ 上的三次埃尔米特插值多项式，其示意曲线如图 2.6 所示。

分段三次埃尔米特插值多项式的余项是

$$R_{H_3}(x)=\frac{f^{(4)}(\xi)}{4!}(x-x_{i-1})^2(x-x_i)^2\quad (\xi\in(x_{i-1},x_i)) \tag{2.58}$$

相应的误差限估计式是

$$|R_{H_3}(x)|=\frac{h^4}{384}\max_{x\in[a,b]}|f^{(4)}(x)|\quad (h=\max_{j=1,\cdots,n}\{h_i\}) \tag{2.59}$$

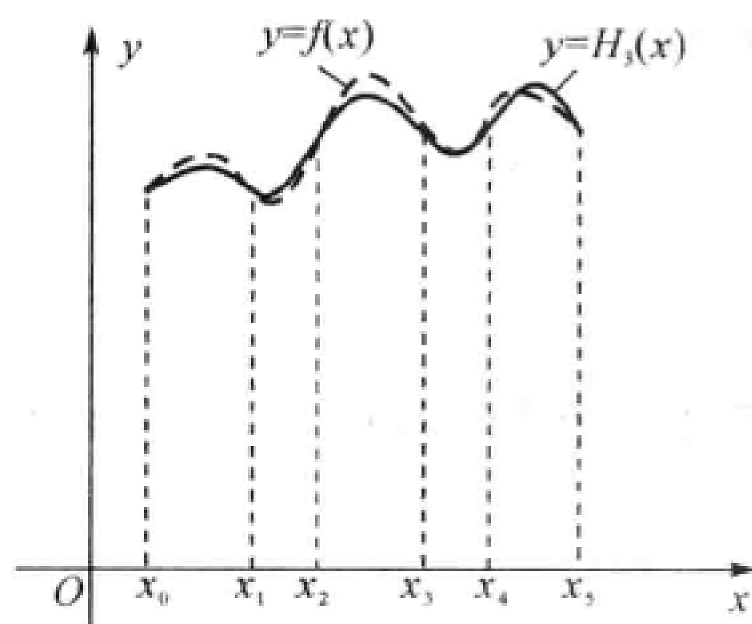


图 2.6

2.4.3 三次样条插值

"样条"原指绘图人员绘制曲线时使用的一种辅助工具，是一个极有弹性的长条。使用时，先把样条固定在一些给定的点(称为"样点")上，并让它自然弯曲，然后依样画下样条呈现的曲线，这样的曲线就称为"样条曲线"。从数学上看，样条曲线是由分段三次多项式曲线构成的光滑曲线，在两个多项式曲线的相接处，不仅多项式的值相等，而且多项式的一阶导数的值和二阶导数的值也分别相等，所以样条曲线光滑性很好。

定义(2.12) 对于给定的插值节点 $x_i(i=0,1,\cdots,n)$，如果函数 $S(x)$ 在每一个小区间 $[x_{i-1},x_i]\ (i=1,2,\cdots,n)$ 上是三次多项式，并且在所有内节点 $x_i(i=1,2,\cdots,n-1)$ 处有直到二阶的连续导数，那么就称 $S(x)$ 为节点 $x_i(i=0,1,\cdots,n)$ 上的三次样条函数。

定义(2.13) 若三次样条函数 $S(x)$ 在所有节点处满足插值条件，即

$$S(x_i)=f(x_i)\quad (i=0,1,\cdots,n) \tag{2.60}$$

则称 $S(x)$ 为三次样条插值函数。

显然，三次样条插值函数就是通过全部样点(即插值点)的二阶连续可微的分段三次多项式函数。设

$$S(x)=S_i(x)\quad (x\in[x_{i-1},x_i],i=1,2,\cdots,n) \tag{2.61}$$

式中

$$S_i(x)=a_ix^3+b_ix^2+c_ix+d_i \quad (i=1,2,\cdots,n) \tag{2.62}$$

三次样条函数 $S(x)$ 中包含 $4n$ 个待定常数 $\{a_i,b_i,c_i,d_i\}$ $(i=1,2,\cdots,n)$。它们可以由下面的三类条件来确定：

(1) $S(x)$，$S'(x)$，$S''(x)$ 在 $n-1$ 个内节点处连续，即

$$\left.\begin{aligned} &S(x_i-0)=S(x_i+0)\\ &S'(x_i-0)=S'(x_i+0) \quad (i=1,2,\cdots,n-1)\\ &S''(x_i-0)=S''(x_i+0)\end{aligned}\right\} \tag{2.63}$$

(2) $S(x)$ 满足插值条件式(2.60)。

以上共有 $3(n-1)+(n+1)=(4n-2)$ 个条件，完全确定样条函数所需的另外两个条件由边界条件给出。

(3) 边界条件就是样条函数在插值区间端点 x_0 和 x_n 处满足的条件。常见的边界条件有三种：

第一种边界条件：给定端点处函数的一阶导数值，即

$$S'(x_0)=f'(x_0),\quad S'(x_n)=f'(x_n) \tag{2.64}$$

第二种边界条件：给定端点处函数的二阶导数值，即

$$S''(x_0)=f''(x_0),\quad S''(x_n)=f''(x_n) \tag{2.65}$$

作为特例，条件 $S''(x_0)=S''(x_n)=0$ 称为自然边界，满足自然边界的样条函数称为自然样条，它是满足插值条件的最光滑的三次样条插值函数。

第三种边界条件：当被插值函数是以 (x_n-x_0) 为周期的周期函数时，样条函数 $S(x)$ 必须满足周期条件

$$\left.\begin{aligned} &S(x_0+0)=S(x_n-0)\\ &S'(x_0+0)=S'(x_n-0)\\ &S''(x_0+0)=S''(x_n-0)\end{aligned}\right\} \tag{2.66}$$

真正起作用的是后两个等式。

补充了两个端点条件后，可以建立 $4n$ 个待定常数 $\{a_i,b_i,c_i,d_i\}$ $(i=1,2,\cdots,n)$ 满足的方程组，求解方程组就能确定三次样条插值函数 $S(x)$。

定理(2.6)　三次样条插值问题的解存在且唯一。

求解三次样条插值函数的常用方法有三弯矩法和三转角法。这里只给出三弯矩法。

在区间 $[x_{i-1},x_i]$ 上满足条件式(2.60)和式(2.63)的 $S_i(x)$ 是

$$\begin{aligned} S_i(x)=&\frac{(x_i-x)^3}{6h_i}M_{i-1}+\frac{(x-x_{i-1})^3}{6h_i}M_i+\left(f(x_{i-1})-\frac{M_{i-1}h_i^2}{6}\right)\frac{x_i-x}{h_i}+\\ &\left(f(x_i)-\frac{M_ih_i^2}{6}\right)\frac{x-x_{i-1}}{h_i} \quad (i=1,2,\cdots,n)\end{aligned} \tag{2.67}$$

其中

$$\left.\begin{aligned} &M_i=S''(x_i) \quad (i=0,1,2,\cdots,n)\\ &h_i=x_i-x_{i-1} \quad (i=1,2,\cdots,n)\end{aligned}\right\} \tag{2.68}$$

由一阶导数在节点 $x_i(i=1,2,\cdots,n-1)$ 处连续的要求 $S'(x_i-0)=S'(x_i+0)$，有

$$\mu_iM_{i-1}+2M_i+\lambda_iM_{i+1}=d_i \quad (i=1,2,\cdots,n-1) \tag{2.69}$$

式中

$$\left.\begin{aligned}&\lambda_i=\frac{h_{i+1}}{h_i+h_{i+1}}\\&\mu_i=1-\lambda_i=\frac{h_i}{h_i+h_{i+1}}\qquad\qquad(i=1,2,\cdots,n-1)\\&d_i=\frac{6}{h_i+h_{i+1}}\left(\frac{f(x_{i+1})-f(x_i)}{h_{i+1}}-\frac{f(x_i)-f(x_{i-1})}{h_i}\right)=6f[x_{i-1},x_i,x_{i+1}]\end{aligned}\right\}\tag{2.70}$$

力学上把 M_i 解释为梁在截面 x_i 处的弯矩，所以公式(2.69) 称为三弯矩方程。三弯矩方程共有 $n-1$ 个，要完全确定 $n+1$ 个 M_i，还须根据边界条件补充两个方程。

对于不同的边界条件，能够建立求解 M_i 的不同线性方程组：

(1) 对于第一种边界条件式(2.64)，求解 M_i 的线性方程组是

$$\begin{bmatrix}2&1&&&&\\\mu_1&2&\lambda_1&&&\\&\mu_2&2&\lambda_2&&\\&&\ddots&\ddots&\ddots&\\&&&\mu_{n-1}&2&\lambda_{n-1}\\&&&&1&2\end{bmatrix}\begin{bmatrix}M_0\\M_1\\\\\vdots\\M_{n-1}\\M_n\end{bmatrix}=\begin{bmatrix}d_0\\d_1\\\\\vdots\\d_{n-1}\\d_n\end{bmatrix}\tag{2.71}$$

其中

$$\left.\begin{aligned}d_0&=2M_0+M_1=\frac{6}{h_1}\left(\frac{f(x_1)-f(x_0)}{h_1}-f'(x_0)\right)\\d_n&=M_{n-1}+2M_n=\frac{6}{h_n}\left(f'(x_n)-\frac{f(x_n)-f(x_{n-1})}{h_n}\right)\end{aligned}\right\}\tag{2.72}$$

(2) 对于第二种边界条件式(2.65)，已知 $M_0=f''(x_0)$，$M_n=f''(x_n)$，求解 $M_i(i=1,2,\cdots,n-1)$ 的线性方程组是

$$\begin{bmatrix}2&\lambda_1&&&&\\\mu_2&2&\lambda_2&&&\\&\mu_3&2&\lambda_3&&\\&&\ddots&\ddots&\ddots&\\&&&\mu_{n-2}&2&\lambda_{n-2}\\&&&&\mu_{n-1}&2\end{bmatrix}\begin{bmatrix}M_1\\M_2\\\\\vdots\\M_{n-2}\\M_{n-1}\end{bmatrix}=\begin{bmatrix}d_1-\mu_1M_0\\d_2\\\\\vdots\\d_{n-2}\\d_{n-1}-\lambda_{n-1}M_n\end{bmatrix}\tag{2.73}$$

(3) 对于第三种边界条件式(2.66)，求解 $M_i(i=1,2,\cdots,n)$ 的线性方程组是

$$\begin{bmatrix}2&\lambda_1&&&&\mu_1\\\mu_2&2&\lambda_2&&&\\&\mu_3&2&\lambda_3&&\\&&\ddots&\ddots&\ddots&\\&&&\mu_{n-1}&2&\lambda_{n-1}\\\lambda_n&&&&\mu_n&2\end{bmatrix}\begin{bmatrix}M_1\\M_2\\\\\vdots\\M_{n-1}\\M_n\end{bmatrix}=\begin{bmatrix}d_1\\d_2\\\\\vdots\\d_{n-1}\\d_n\end{bmatrix}\tag{2.74}$$

式中

$$\left.\begin{aligned}&\lambda_n=\frac{h_1}{h_1+h_n}\\&\mu_n=1-\lambda_n=\frac{h_n}{h_1+h_n}\\&d_n=\frac{6}{h_1+h_n}\left(\frac{f(x_1)-f(x_0)}{h_1}-\frac{f(x_n)-f(x_{n-1})}{h_n}\right)\end{aligned}\right\}\tag{2.75}$$

并且

$$M_0=M_n,\quad d_n=\mu_n M_{n-1}+2M_n+\lambda_n M_1\tag{2.76}$$

总结上述过程,可将建立三次样条插值函数 $S(x)$ 的过程归纳如下:

(1) 根据给定的插值点 $(x_i,f(x_i))$ $(i=0,1,2,\cdots,n)$ 及边界条件,建立求解 M_i $(i=0,1,2,\cdots,n)$ 的线性方程组式(2.71)、式(2.73)或式(2.74)。

(2) 解方程组,求出 M_i $(i=0,1,2,\cdots,n)$。

(3) 将 M_i $(i=0,1,2,\cdots,n)$ 代入公式(2.67),得到 $S(x)$ 在区间 $[x_{i-1},x_i]$ $(i=1,2,\cdots,n)$ 上的分段表达式 $S_i(x)$ $(i=1,2,\cdots,n)$。

(4) 整个插值区间上的三次样条插值函数为

$$S(x)=\begin{cases}S_1(x) & (x\in[x_0,x_1])\\S_2(x) & (x\in[x_1,x_2])\\\vdots & \quad\vdots\\S_n(x) & (x\in[x_{n-1},x_n])\end{cases}\tag{2.77}$$

例 2.5　已知函数的表格形式,见表 2.8。试在区间[1,5]上用三弯矩法给出三次自然样条插值函数。

表　2.8

i	0	1	2	3
x_i	1	2	4	5
$f(x_i)$	1	3	4	2

解　对于自然边界条件,有 $M_0=M_3=0$。根据公式(2.68)和公式(2.70)求得的有关参数是

$$h_1=1,\quad h_2=2,\quad h_3=1,\quad \lambda_1=\frac{2}{3},\quad \mu_2=\frac{2}{3},\quad d_1=-3,\quad d_2=-5$$

代入公式(2.73),有线性方程组

$$\begin{cases}2M_1+\dfrac{2}{3}M_2=-3\\[2ex]\dfrac{2}{3}M_1+2M_2=-5\end{cases}$$

解得

$$M_1=-\frac{3}{4},\quad M_2=-\frac{9}{4}$$

代入公式(2.67),经过整理得到所求的三次自然样条插值函数是

$$S(x)=\begin{cases}-\frac{1}{8}x^3+\frac{3}{8}x^2+\frac{7}{4}x-1 & (1\leqslant x\leqslant 2)\\ -\frac{1}{8}x^3+\frac{3}{8}x^2+\frac{7}{4}x-1 & (2\leqslant x\leqslant 4)\\ \frac{3}{8}x^3-\frac{45}{8}x^2+\frac{103}{4}x-33 & (4\leqslant x\leqslant 5)\end{cases}$$

可以通过增加节点来提高三次样条差值的精确度。对于三次样条插值函数，当插值节点逐渐加密时，不但样条函数收敛于被插值函数，而且样条函数的导数也收敛于被插值函数的导数，这种性质要比多项式插值优越得多。

2.5 函数拟合的最小二乘法

对于许多实际问题来说，函数关系 $y=f(x)$ 中变量的值是通过实验测量得到的，即用测量数据 $(x_i,f(x_i))$ $(i=0,1,2,\cdots,m)$ 给出。测量数据总是有误差的，因此若像插值法那样要求近似函数严格通过每一个测量数据点，则不可避免地就把测量误差引入到函数关系中，难以反映变量之间关系的真实情况。函数拟合就是用简单函数 $\varphi(x)$ 的计算点 $(x_i,\varphi(x_i))$ $(i=0,1,2,\cdots,m)$ 去逼近函数 $y=f(x)$ 的测量点 $(x_i,f(x_i))$ $(i=0,1,2,\cdots,m)$，也就是用函数 $\varphi(x)$ 去逼近函数 $f(x)$，逼近的标准是函数 $f(x)$ 和 $\varphi(x)$ 在 x_i $(i=0,1,2,\cdots,m)$ 处的整体偏差最小，并不要求完全相等。这样构造的逼近函数 $\varphi(x)$ 称为拟合函数。

定义(2.14) 对于给定的一组数据 $(x_i,f(x_i))$ $(i=0,1,2,\cdots,m)$，在某一简单函数类 Φ 中寻找一个函数 $\varphi^*(x)$，使得

$$R=\sum_{i=0}^{m}|\varphi^*(x_i)-f(x_i)|^2=\min_{\varphi\in\Phi}\sum_{i=0}^{m}|\varphi(x_i)-f(x_i)|^2 \tag{2.78}$$

这种方法称为函数拟合的最小二乘法。上式称为最小二乘原则，按照最小二乘原则确定的拟合曲线 $y=\varphi^*(x)$ 称为最小二乘拟合曲线。

如图 2.7 所示，曲线拟合的最小二乘法就是用简单函数 $y=\varphi(x)$ 曲线去逼近已知数据点 $(x_i,f(x_i))$ $(i=0,1,2,\cdots,m)$，不要求曲线通过所有数据点，只要求函数曲线与数据点在整体上最为靠近，即均方误差 $\sqrt{\sum_{i=0}^{m}|\varphi(x_i)-f(x_i)|^2}$ 最小。

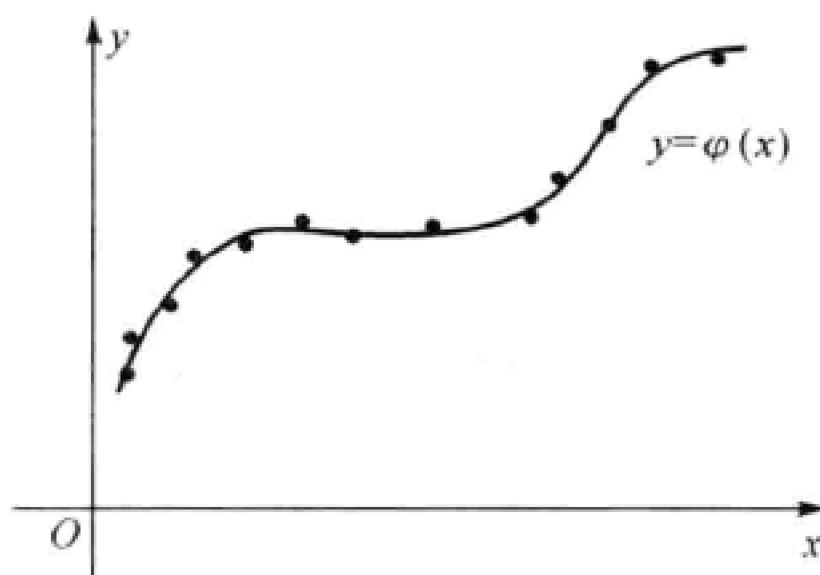


图 2.7

2.5.1 多项式拟合

取 Φ 为代数多项式的函数拟合称为多项式拟合。这里只介绍多项式最小二乘法函数拟合。

设拟合函数为 n 次多项式

$$P_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n \tag{2.79}$$

则

$$R(a_0,a_1,\cdots,a_n)=\sum_{i=0}^{m}|P_n(x_i)-f(x_i)|^2$$

可以把 $R(a_0,a_1,\cdots,a_n)$ 看作 $a_i(i=0,1,\cdots,n)$ 的函数($n\leqslant m$)，即

$$R(a_0,a_1,\cdots,a_n)=\sum_{i=0}^{m}|a_0+a_1x_i+a_2x_i^2+\cdots+a_nx_i^n-f(x_i)|^2 \tag{2.80}$$

多项式最小二乘法函数拟合就是，对于给定数据 $(x_i,f(x_i))$ $(i=0,1,2,\cdots,m)$ 和 n 次多项式(2.79)，寻找一组 $a_i(i=0,1,\cdots,n)$，使式(2.80)的值最小。根据多元函数取极值的必要条件可知，$R(a_0,a_1,\cdots,a_n)$ 必须满足

$$\frac{\partial R}{\partial a_i}=0 \quad (i=0,1,\cdots,n) \tag{2.81}$$

得到 $a_i(i=0,1,\cdots,n)$ 应该满足的方程组

$$\left.\begin{aligned}
&a_0(m+1)+a_1\sum_{i=0}^{m}x_i+a_2\sum_{i=0}^{m}x_i^2+\cdots+a_n\sum_{i=0}^{m}x_i^n=\sum_{i=0}^{m}f(x_i)\\
&a_0\sum_{i=0}^{m}x_i+a_1\sum_{i=0}^{m}x_i^2+a_2\sum_{i=0}^{m}x_i^3+\cdots+a_n\sum_{i=0}^{m}x_i^{n+1}=\sum_{i=0}^{m}x_if(x_i)\\
&\qquad\vdots\\
&a_0\sum_{i=0}^{m}x_i^n+a_1\sum_{i=0}^{m}x_i^{n+1}+a_2\sum_{i=0}^{m}x_i^{n+2}+\cdots+a_n\sum_{i=0}^{m}x_i^{2n}=\sum_{i=0}^{m}x_i^nf(x_i)
\end{aligned}\right\} \tag{2.82}$$

它被称为法方程或正规方程，写成矩阵形式是

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\boldsymbol{\alpha}=\mathbf{A}^{\mathrm{T}}\mathbf{y} \tag{2.83}$$

式中

$$\mathbf{A}=\begin{bmatrix}1 & x_0 & x_0^2 & \cdots & x_0^n\\ 1 & x_1 & x_1^2 & \cdots & x_1^n\\ \vdots & \vdots & \vdots & & \vdots\\ 1 & x_m & x_m^2 & \cdots & x_m^n\end{bmatrix},\quad \boldsymbol{\alpha}=\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_n\end{bmatrix},\quad \mathbf{y}=\begin{bmatrix}f(x_0)\\ f(x_1)\\ \vdots\\ f(x_m)\end{bmatrix} \tag{2.84}$$

解正规方程式(2.82)得出 $a_i(i=0,1,\cdots,n)$，然后代入式(2.79)就得到拟合多项式。由此得到的函数曲线 $P_n(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n$ 称为已知数据 $(x_i,f(x_i))$ $(i=0,1,2,\cdots,m)$ 的经验曲线。

对于线性拟合问题，拟合函数为一次多项式

$$P_1(x)=a_0+a_1x \tag{2.85}$$

多项式系数 a_0 和 a_1 满足的法方程式(2.83)中的几个矩阵分别是

$$\mathbf{A}=\begin{bmatrix}1 & x_0\\ 1 & x_1\\ \vdots & \vdots\\ 1 & x_m\end{bmatrix},\quad \boldsymbol{\alpha}=\begin{bmatrix}a_0\\ a_1\end{bmatrix},\quad \mathbf{y}=\begin{bmatrix}f(x_0)\\ f(x_1)\\ \vdots\\ f(x_m)\end{bmatrix} \tag{2.86}$$

法方程式(2.83)也可以表示为

$$\begin{bmatrix}1 & \langle x\rangle\\ \langle x\rangle & \langle x^2\rangle\end{bmatrix}\begin{bmatrix}a_0\\ a_1\end{bmatrix}=\begin{bmatrix}\langle y\rangle\\ \langle xy\rangle\end{bmatrix} \tag{2.87}$$

其中的几个平均值分别是

$$\langle x \rangle = \frac{1}{m+1}\sum_{i=0}^{m} x_i, \qquad \langle x^2 \rangle = \frac{1}{m+1}\sum_{i=0}^{m} x_i^2$$
$$\langle y \rangle = \frac{1}{m+1}\sum_{i=0}^{m} f(x_i), \quad \langle xy \rangle = \frac{1}{m+1}\sum_{i=0}^{m} x_i f(x_i) \tag{2.88}$$

方程式(2.87)的解是

$$a_1 = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2}, \quad a_0 = \langle y \rangle - a_1 \langle x \rangle \tag{2.89}$$

由此可以确定给定数据$(x_i, f(x_i))$ $(i=0,1,2,\cdots,m)$的经验直线$y=a_0+a_1x$，其均方误差是

$$Q = \sqrt{\sum_{i=0}^{m} [a_0 + a_1 x_i - f(x_i)]^2}$$

例 2.6 有一组实验数据，见表 2.9。试用最小二乘法求经验直线 $y=a_0+a_1x$ 及其均方误差。

表 2.9

x_i	1	2	3	4	5
$f(x_i)$	4	4.5	6	8	8.5

解 对于本问题，公式(2.86)中的几个矩阵分别是

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix}, \quad \alpha = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 4 \\ 4.5 \\ 6 \\ 8 \\ 8.5 \end{bmatrix}$$

代入法方程式(2.87)，就是

$$\begin{bmatrix} 1 & 3 \\ 3 & 11 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 6.2 \\ 21.1 \end{bmatrix}$$

其解为

$$a_0 = 2.45, \quad a_1 = 1.25$$

经验直线公式是

$$y = 2.45 + 1.25x$$

均方误差为

$$Q = \sqrt{\sum_{i=0}^{4} [(2.45 + 1.25x_i) - f(x_i)]^2} = 0.835$$

进行函数拟合时，通常先将数据$(x_i, f(x_i))$ $(i=0,1,2,\cdots,m)$绘成函数曲线$f(x) \sim x$，然后分析曲线形状，从而确定要选择的函数类Φ。有些实验数据在坐标纸上描出的点不呈现直线分布，但经过变量代换后可以转化为直线分布，如表 2.10 给出的几种情况。

表　2.10

函数关系 $y=f(x)$	线性化形式 $Y=CX+D$	转化关系
$y=\frac{A}{x}+B$	$y=A\frac{1}{x}+B$	$Y=y,X=\frac{1}{x},C=A,D=B$
$y=\frac{A}{x+B}$	$y=\frac{-1}{B}(xy)+\frac{A}{B}$	$Y=y,X=xy,C=-\frac{1}{B},D=\frac{A}{B}$
	$\frac{1}{y}=\frac{1}{A}x+\frac{B}{A}$	$Y=\frac{1}{y},X=x,C=\frac{1}{A},D=\frac{B}{A}$
$y=\frac{1}{Ax+B}$	$\frac{1}{y}=Ax+B$	$Y=\frac{1}{y},X=x,C=A,D=B$
$y=\frac{x}{Ax+B}$	$\frac{1}{y}=B\frac{1}{x}+A$	$Y=\frac{1}{y},X=\frac{1}{x},C=B,D=A$
$y=A\ln x+B$	$y=A\ln x+B$	$Y=y,X=\ln x,C=A,D=B$
$y=Be^{Ax}$	$\ln y=Ax+\ln B$	$Y=\ln y,X=x,C=A,D=\ln B$
$y=Bx^{A}$	$\ln y=A\ln x+\ln B$	$Y=\ln y,X=\ln x,C=A,D=\ln B$
$y=\frac{1}{(Ax+B)^2}$	$\frac{1}{y^{1/2}}=Ax+B$	$Y=\frac{1}{y^{1/2}},X=x,C=A,D=B$
$y=Bxe^{-Ax}$	$\ln\frac{y}{x}=-Ax+\ln B$	$Y=\ln\frac{y}{x},X=x,C=-A,D=\ln B$
$y=\frac{1}{1+Be^{Ax}}$	$\ln\left(\frac{1}{y}-1\right)=Ax+\ln B$	$Y=\ln\left(\frac{1}{y}-1\right),X=x,C=A,D=\ln B$

例 2.7　给定数据$(x_i,y_i)(i=0,1,2,3,4)$，见表 2.11。试用最小二乘法确定变量 y 与 x 的函数关系。

表　2.11

x_i	1.00	1.25	1.50	1.75	2.00
y_i	5.10	5.79	6.53	7.45	8.46

解　题中没有指明两个变量的函数关系类型，这时需要通过画图并观察测量点的分布特点，以确定拟合曲线类型。由图 2.8 可见，数据分布接近一指数曲线，因此选择拟合函数为

$$y=ae^{bx}$$

a,b 为待定常数。这是一个非线性函数，先将其转化为线性函数。为此对函数两边取对数，得

$$\ln y=\ln a+bx$$

令

$$z=\ln y,\quad A=\ln a$$

于是有

$$z=A+bx$$

这是一个线性函数,可以用最小二乘法求解。

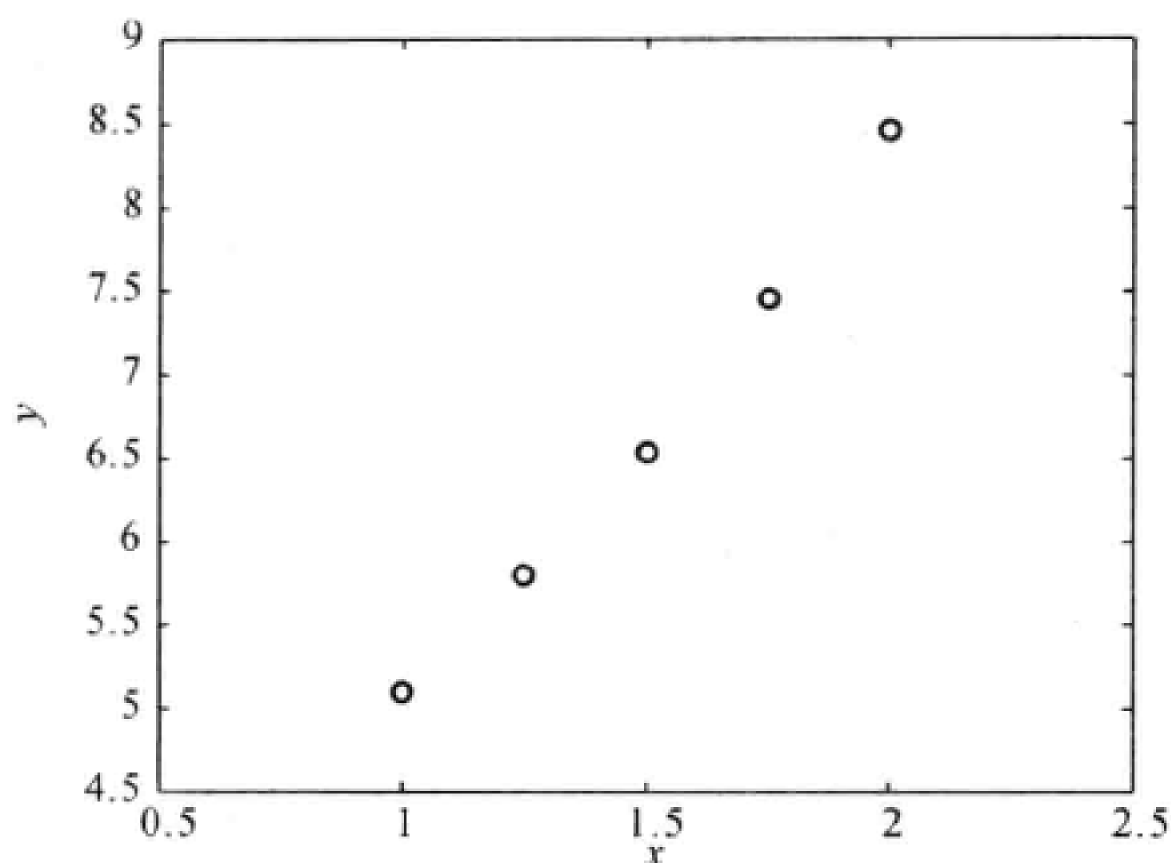


图 2.8

将所给数据转化为(x_i,z_i) $(i=0,1,2,3,4)$,见表2.12。

表 2.12

x_i	1.00	1.25	1.50	1.75	2.00
z_i	1.629	1.756	1.876	2.008	2.135

得法方程为

$$\begin{bmatrix} 5 & 7.50 \\ 7.50 & 11.875 \end{bmatrix} \begin{bmatrix} A \\ b \end{bmatrix} = \begin{bmatrix} 9.404 \\ 14.422 \end{bmatrix}$$

解得

$$A=1.122, \quad b=0.5056$$

因此 $a=e^A=3.071$,于是最小二乘拟合函数是

$$y=3.071e^{0.5056x}$$

程序(2.6) 最小二乘法线性拟合的 MATLAB 程序。

程序任务:根据数据点(x_i,y_i) $(i=1,2,\cdots,n)$构造最小二乘法拟合直线 $y=ax+b$。

```
function [a, b] = lsline(x, y)
% 输入:x—— 由坐标点 x_i (i = 1,2,…,n) 组成的行向量
%      y—— 由函数值 y_i (i = 1,2,…,n) 组成的行向量
% 输出:a—— 方程 y = ax + b 中变量 x 前的系数
%      b—— 方程 y = ax + b 中的常系数
xm = mean(x);                   % 计算〈x〉
ym = mean(y);                   % 计算〈y〉
s2 = (x - xm) * (x - xm) ';     % 计算 n(〈x²〉 - 〈x〉²)
sy = (y - ym) * (x - xm) ';     % 计算 n(〈xy〉 - 〈x〉〈y〉)
a = sy/s2;                      %a = (〈xy〉 - 〈x〉〈y〉) / (〈x²〉 - 〈x〉²)
b = ym - a * xm;                %b = 〈y〉 - a〈x〉
```

在 MATLAB 命令窗口输入：

```
>> x = -2:2;                   % 输入数据点 x 坐标向量
>> y = [1,2,3,3,4];            % 输入数据点 y 坐标向量
>> [a,b] = lsline(x, y);       % 调用程序(2.6) 计算拟合直线 y = ax + b 中的常数
```

输出结果：

a = 0.7000，　b = 2.6000

即拟合直线方程是 $y = 0.7x + 2.6$

```
>> x1 = -2.5:0.1:2;            % 以下绘制数据点和拟合直线
>> y1 = a.* x1 + b;
>> plot(x, y, 'ok', x1, y1, '-k');
```

输出结果如图 2.9 所示(图中实线表示拟合直线,圆圈代表数据点)。

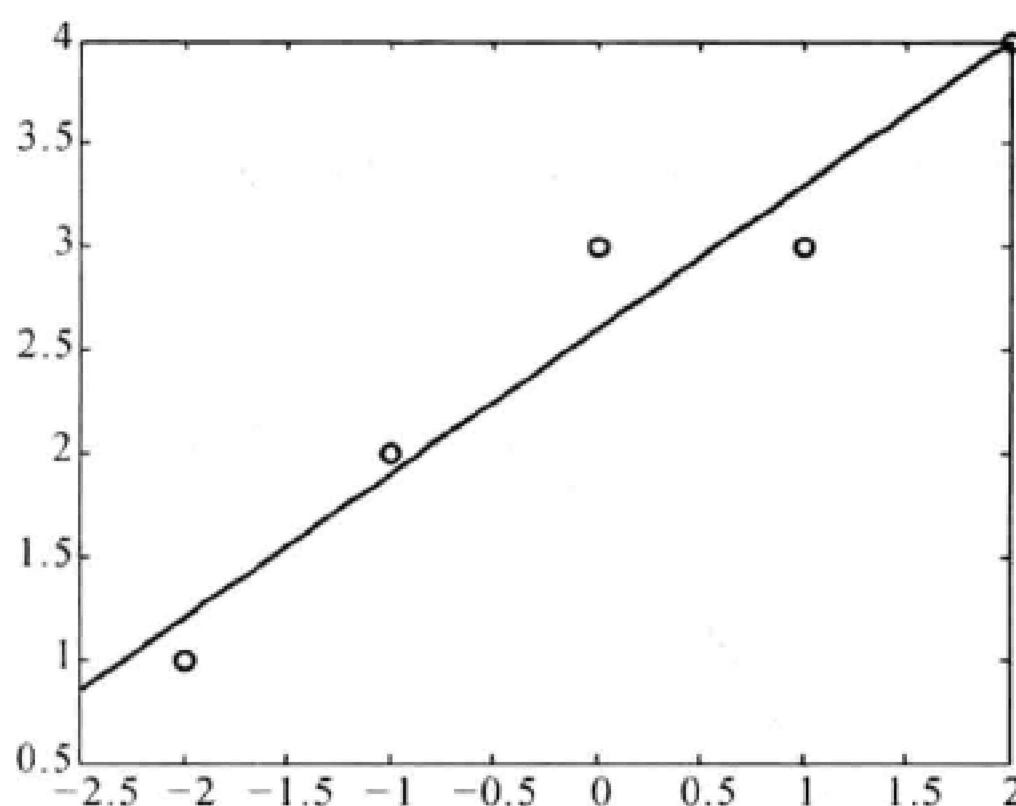


图　2.9

程序(2.7)　最小二乘法 m 阶多项式拟合的 MATLAB 程序。

程序任务：根据 n 个数据点 (x_i, y_i) $(i = 1, 2, \cdots, n)$ 拟合 m 阶多项式。

```
function c = lspoly(x,y,m)
% 输入：x—— 由坐标点 x_i (i = 1,2,…,n) 组成的行向量
%       y—— 由函数值 y_i (i = 1,2,…,n) 组成的行向量
%       m—— 最小二乘拟合多项式的阶
% 输出：c—— 建立的多项式系数向量
n = length(x);
B = zeros(1:m+1);
F = zeros(n, m+1);
% 用 x 的不同次幂构造矩阵 F
for k = 1:m+1
    F(:,k) = x'.^(k-1);
end
% 解线性方程组 F'Fa = F'y'
A = F' * F;
B = F' * y';
```

```
a = A\B;          % 解线性方程组 Aa = B
c = flipud(a);    % 上下翻转向量 a
```

在 MATLAB 命令窗口输入：

```
>> x = -2:2;                              % 输入数据点 x 坐标向量
>> y = [-5.8,1.1,3.8,3.3,-1.5];           % 输入数据点 y 坐标向量
>> c = lspoly(x,y,2);                     % 调用程序(2.7) 拟合二次多项式
```

输出结果：

c =[-1.9000,1.0800,3.9800]'

即拟合曲线方程是 $y=-1.9x^2+1.08x+3.98$

```
>> x1 = -2:0.1:2;                         % 以下绘制数据点和拟合曲线
>> y1 = polyval(c,x1);
>> plot(x,y,'ok',x1,y1,'-k');
```

输出结果如图 2.10 所示(图中实线表示拟合曲线,圆圈代表数据点))。

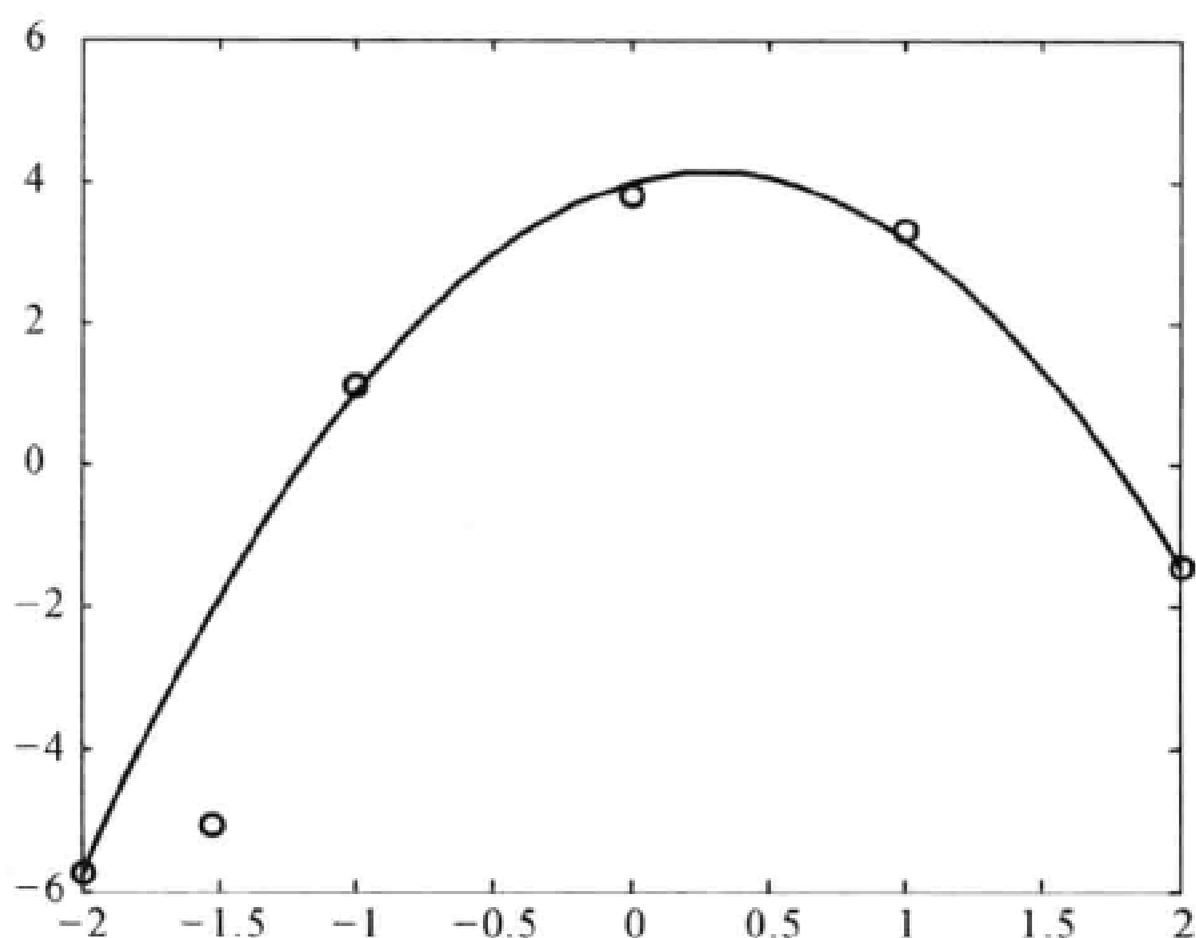


图 2.10

2.5.2 多元线性拟合

设变量 y 是多个变量 $x_j(j=1,2,\cdots,k)$ 的函数。现测得 n 组数据 $(x_{1i},x_{2i},\cdots,x_{ki};y_i)(i=1,2,\cdots,n)$，一般应满足 $n>k$。作为一级近似，假设变量 Y 是变量 $x_j(j=1,2,\cdots,k)$ 的线性函数，即

$$Y=A_0+A_1x_1+A_2x_2+\cdots+A_kx_k \tag{2.90}$$

与函数拟合的最小二乘法思路相同，将测量数据代入上式，得到 n 个关于 $A_0,A_1,\cdots,A_k$ 的方程

$$Y_i=A_0+A_1x_{1i}+A_2x_{2i}+\cdots+A_kx_{ki}\quad(i=1,2,\cdots,n)$$

用最小二乘原理确定 $A_0,A_1,\cdots,A_k$，使 y_i 与 Y_i 偏差的平方和最小，即

$$R=\sum_{i=1}^{n}(y_i-A_0-A_1x_{1i}-A_2x_{2i}-\cdots-A_kx_{ki})^2 \tag{2.91}$$

最小。令式(2.91)对 $A_0, A_1, \cdots, A_k$ 的偏导数等于零并化简，就有正规方程组

$$\left.\begin{aligned}&A_0 = \langle y\rangle - A_1\langle x_1\rangle - A_2\langle x_2\rangle - \cdots - A_k\langle x_k\rangle\\&l_{11}A_1 + l_{12}A_2 + \cdots + l_{1k}A_k = l_{1y}\\&l_{21}A_1 + l_{22}A_2 + \cdots + l_{2k}A_k = l_{2y}\\&\cdots\cdots\\&l_{k1}A_1 + l_{k2}A_2 + \cdots + l_{kk}A_k = l_{ky}\end{aligned}\right\}\tag{2.92}$$

式中

$$l_{rs} = l_{sr} = \langle x_r x_s\rangle - \langle x_r\rangle\langle x_s\rangle,\quad l_{ry} = \langle x_r y\rangle - \langle x_r\rangle\langle y\rangle\qquad (r,s = 1,2,\cdots,k)\tag{2.93}$$

其中

$$\left.\begin{aligned}&\langle x_r x_s\rangle = \frac{1}{n}\sum_{i=1}^{n} x_{ri}x_{si},\quad \langle x_r y\rangle = \frac{1}{n}\sum_{i=1}^{n} x_{ri}y_i\\&\langle x_r\rangle = \frac{1}{n}\sum_{i=1}^{n} x_{ri},\qquad \langle y\rangle = \frac{1}{n}\sum_{i=1}^{n} y_i\end{aligned}\quad (r,s = 1,2,\cdots,k)\right\}\tag{2.94}$$

从方程式(2.92)解出 $A_0, A_1, \cdots, A_k$，就能确定拟合函数式(2.90)。

2.6 快速傅里叶变换

2.6.1 傅里叶变换与离散傅里叶变换

定义(2.15)　设函数 $f(t)$ 定义在 $(-\infty,\infty)$，且满足 $\int_{-\infty}^{\infty}|f(t)|\mathrm{d}t < \infty$，则称

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\,\mathrm{e}^{-2\pi\mathrm{i}\omega t}\,\mathrm{d}t\quad (\mathrm{i} = \sqrt{-1};\ \omega \in (-\infty,\infty))\tag{2.95}$$

是函数 $f(t)$ 的傅里叶(Fourier)变换。可以证明

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\,\mathrm{e}^{2\pi\mathrm{i}t\omega}\,\mathrm{d}\omega\tag{2.96}$$

并称其是函数 $F(\omega)$ 的傅里叶逆变换。

通常，能够获得函数 $f(t)$ 在 $(2N+1)$ 个离散点 $t = k\Delta t(k = 0, \pm 1, \pm 2, \cdots, \pm N; \Delta t > 0)$ 处的值 $f(k\Delta t)$，并据此评估函数 $f(t)$ 的特性，这里分析其可行性。当 N 足够大时，傅里叶变换式(2.95)可以表示为

$$F(\omega) \approx \int_{-N\Delta t}^{N\Delta t} f(t)\,\mathrm{e}^{-2\pi\mathrm{i}\omega t}\,\mathrm{d}t$$

如果同时 Δt 充分小，则上式还能写成

$$F(\omega) \approx \sum_{k=-N}^{N-1} f(k\Delta t)\,\mathrm{e}^{-2\pi\mathrm{i}\omega k\Delta t}\Delta t = \sum_{k=-N}^{-1} f(k\Delta t)\,\mathrm{e}^{-2\pi\mathrm{i}\omega k\Delta t}\Delta t + \sum_{k=0}^{N-1} f(k\Delta t)\,\mathrm{e}^{-2\pi\mathrm{i}\omega k\Delta t}\Delta t =$$
$$\sum_{k=0}^{N-1} f((k-N)\Delta t)\,\mathrm{e}^{-2\pi\mathrm{i}\omega k\Delta t}\mathrm{e}^{2\pi\mathrm{i}\omega N\Delta t}\Delta t + \sum_{k=0}^{N-1} f(k\Delta t)\,\mathrm{e}^{-2\pi\mathrm{i}\omega k\Delta t}\Delta t$$

若取 $\omega = j/(N\Delta t)$ $(j = 0,1,2,\cdots,N-1)$，得

$$F\left(\frac{j}{N\Delta t}\right) \approx \sum_{k=0}^{N-1} A_k\,\mathrm{e}^{-2\pi\mathrm{i}kj/N}$$

其中，$A_k=(f(k\Delta t)+f((k-N)\Delta t))\Delta t$。因此，计算 $F(\omega)$ 在点 $\omega=j/(N\Delta t)(j=0,1,2,\cdots,N-1)$ 处的值就变成了计算求和

$$F_j=\sum_{k=0}^{N-1}A_k\mathrm{e}^{-2\pi\mathrm{i}kj/N}\quad(j=0,1,2,\cdots,N-1)$$

不难推出，其逆运算是

$$A_k=\frac{1}{N}\sum_{j=0}^{N-1}F_j\mathrm{e}^{2\pi\mathrm{i}jk/N}\quad(k=0,1,2,\cdots,N-1)$$

定义(2.16)　序列 $\{a_j\}\Big|_{j=0}^{N-1}$ 的离散傅里叶变换(DFT)是序列 $\{c_k\}\Big|_{k=0}^{N-1}$，其中

$$c_k=\sum_{j=0}^{N-1}a_j\mathrm{e}^{-2\pi\mathrm{i}kj/N}\tag{2.97}$$

而序列 $\{c_k\}\Big|_{k=0}^{N-1}$ 的离散傅里叶逆变换(IDFT)是序列 $\{a_j\}\Big|_{j=0}^{N-1}$，其中

$$a_j=\frac{1}{N}\sum_{k=0}^{N-1}c_k\mathrm{e}^{2\pi\mathrm{i}jk/N}\tag{2.98}$$

记作

$$a_j\Leftrightarrow c_k\tag{2.99}$$

离散傅里叶逆变换把数据 $\{a_j\}\Big|_{j=0}^{N-1}$ 表示为它的基本频率的组合，组合时的系数就是变换后的数据 $\{c_k\}\Big|_{k=0}^{N-1}$。由于 $\exp(2\pi\mathrm{i}jk/N)=\exp(\mathrm{i}(2\pi jk/N))$，因此 jk/N 相当于频率，$2\pi jk/N$ 相当于圆频率。与连续傅里叶变换相比，如果把离散数据 $\{a_j\}\Big|_{j=0}^{N-1}$ 看成是从连续信号中以时间间隔 Δt 取样所得，则第 j 个数据的取样时刻是 $j\Delta t$，全部数据的取样时间长度是 $N\Delta t$，它相当于一个取样周期，所以基频为 $1/N\Delta t$，倍频为 $k/N\Delta t$，把 $t=j\Delta t$ 和 $\nu=k/N\Delta t$ 代入 $\exp(2\pi\mathrm{i}\nu t)$ 就得到 $\exp(2\pi\mathrm{i}jk/N)$。另外，c_k/N 相当于各频率的振幅，把振幅模的二次方称为功率谱。

现在考察离散傅里叶变换与插值多项式系数的关系。设已知函数 $x(t)$ 在区间 $[0,2\pi]$ 内 N 个分立点 $2\pi j/N(j=0,1,2,\cdots,N-1)$ 处的值 $x(j)=x(2\pi j/N)$，用函数 $\mathrm{e}^{\mathrm{i}kt}(k=0,1,2,\cdots,N-1)$ 的线性组合

$$P(t)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\mathrm{e}^{\mathrm{i}kt}\tag{2.100}$$

作为函数 $x(t)$ 在区间 $[0,2\pi]$ 上的插值函数。显然，确定系数 $X(k)\ (k=0,1,2,\cdots,N-1)$ 的条件是函数 $P(t)$ 和 $x(t)$ 在 N 个分立点 $2\pi j/N(j=0,1,2,\cdots,N-1)$ 处的值相等，即

$$x(j)=P\left(\frac{2\pi j}{N}\right)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\mathrm{e}^{2\pi\mathrm{i}jk/N}\quad(j=0,1,2,\cdots,N-1)$$

从中可以解出

$$X(k)=\sum_{j=0}^{N-1}x(j)\mathrm{e}^{-2\pi\mathrm{i}kj/N}\quad(k=0,1,2,\cdots,N-1)\tag{2.101}$$

式(2.101)表明，函数 $x(t)$ 关于离散点 $x(j)=x(2\pi j/N)\ (j=0,1,2,\cdots,N-1)$ 插值多项式(2.100)的系数 $X(k)\ (k=0,1,2,\cdots,N-1)$ 就是函数 $x(t)$ 在离散点的值序列 $\{x(j)\}\Big|_{j=0}^{N-1}$ 的

离散傅里叶变换。

若记

$$\omega = e^{-2\pi i/N} \quad (N \text{ 是正整数}) \tag{2.102}$$

称 $N \times N$ 矩阵

$$\boldsymbol{F} = [f_{kj}] = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{N-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix} \tag{2.103}$$

为 N 阶傅里叶矩阵，其中

$$f_{kj} = \omega^{kj} = e^{-2\pi ikj/N} \tag{2.104}$$

对于给定的 N 维向量

$$\boldsymbol{a} = [a_0 \quad a_1 \quad \cdots \quad a_{N-1}]^{\mathrm{T}} \tag{2.105}$$

向量

$$\boldsymbol{c} = [c_0 \quad c_1 \quad \cdots \quad c_{N-1}]^{\mathrm{T}} = \boldsymbol{Fa} \tag{2.106}$$

就是向量 $\boldsymbol{a}$ 的离散傅里叶变换；而向量

$$\boldsymbol{a} = \boldsymbol{F}^{-1}\boldsymbol{c} \tag{2.107}$$

是向量 $\boldsymbol{c}$ 的离散傅里叶逆变换。

考虑到 ω 的特点，可以发现傅里叶矩阵 $\boldsymbol{F}$ 是一个元素为复数的对称范德蒙矩阵，并且可以证明

$$\boldsymbol{F}^{-1} = \frac{1}{N}\boldsymbol{F}^{*} = \frac{1}{N}\bar{\boldsymbol{F}} \tag{2.108}$$

式中，$\boldsymbol{F}^{*}$ 和 $\bar{\boldsymbol{F}}$ 分别是矩阵 $\boldsymbol{F}$ 的复共轭矩阵和复共轭转置矩阵。

设

$$p(x) = \sum_{j=0}^{N-1} a_j x^j \tag{2.109}$$

对比式(2.97)，有

$$c_k = p(\omega^k) \quad (k = 0, 1, 2, \cdots, N-1) \tag{2.110}$$

式(2.110) 表明，离散傅里叶变换等价于计算多项式 $p(x)$ 在 $\omega^k (k=0,1,2,\cdots,N-1)$ 处的值；而离散傅里叶逆变换相当于计算多项式 $p(x)$ 的系数 $a_j (j=0,1,2,\cdots,N-1)$，使得式(2.109) 成立。

离散傅里叶变换 $\boldsymbol{c} = \boldsymbol{Fa}$ 就是矩阵乘法。但如果直接用矩阵乘法计算离散傅里叶变换，则当 N 比较大时，计算量相当可观。若傅里叶矩阵已知，每计算向量 $\boldsymbol{c}$ 的一个元素，就要进行 N 次乘法运算和 $N-1$ 次加法运算，离散傅里叶变换的运算总次数就是 $(2N-1)N \approx 2N^2$。因此研究离散傅里叶变换的快速算法十分必要。

2.6.2 快速傅里叶变换

1965 年 Cooly 和 Tuley 提出了适合在计算机上使用的离散傅里叶变换快速计算方法——快速傅里叶变换(FFT)，显著提高了运算速度，使傅里叶变换得到更广泛的应用。快速傅里叶

变换算法的核心思想是尽量减少乘法次数。式(2.104)中的 $f_{kj}=\omega^{kj}$ 共有 N^2 个元素，但这 N^2 个元素实际上只有 N 个不同的值 $\omega^k(k=0,1,2,\cdots,N-1)$。计算时，可以先把式(2.97)中的 a_j 各项按 $\omega^k(k=0,1,2,\cdots,N-1)$ 归类，然后把同类项中的 a_j 加起来之后再与 ω^k 相乘，这样就能大量减少乘法运算次数。

若 N 是偶数，则多项式(2.109)重新表示为

$$
\begin{aligned}
p(x)&=a_0+a_1x+a_2x^2+\cdots+a_{N-1}x^{N-1}=\\
&(a_0+a_2x^2+a_4x^4+\cdots+a_{N-2}x^{N-2})+x(a_1+a_3x^2+a_5x^4+\cdots+a_{N-1}x^{N-2})=\\
&p_e(x^2)+xp_o(x^2)
\end{aligned}
\tag{2.111}
$$

其中

$$p_e(t)=a_0+a_2t+a_4t^2+\cdots+a_{N-2}t^{N/2-1}=\sum_{r=0}^{N/2-1}a_{2r}t^r \tag{2.112}$$

$$p_o(t)=a_1+a_3t+a_5t^2+\cdots+a_{N-1}t^{N/2-1}=\sum_{r=0}^{N/2-1}a_{2r+1}t^r \tag{2.113}$$

引入

$$\omega_n=\mathrm{e}^{-2\pi\mathrm{i}/n}\quad(n\text{ 是自然数}) \tag{2.114}$$

则有

$$\omega_N^{2j}=\mathrm{e}^{-j[2\pi\mathrm{i}/(N/2)]}=\omega_{N/2}^j,\omega_N^{j+N/2}=\mathrm{e}^{-j[2\pi\mathrm{i}/N]}\mathrm{e}^{-\pi\mathrm{i}}=-\omega_N^j\quad(j=0,1,2,\cdots,N/2-1) \tag{2.115}$$

这样，从公式(2.110)和公式(2.111)可得，离散傅里叶变换向量的分量是

$$c_k=v_k+\omega_N^k u_k,\quad c_{k+N/2}=v_k-\omega_N^k u_k\quad(k=0,1,2,\cdots,N/2-1) \tag{2.116}$$

式中

$$v_k=p_e(\omega_{N/2}^k),\quad u_k=p_o(\omega_{N/2}^k) \tag{2.117}$$

因此，计算 $p(x)$ 在 $\omega_N^0,\omega_N^1,\omega_N^2,\cdots,\omega_N^{N-1}$ 这 N 个点上的值的问题(即 N 个数据构成序列的离散傅里叶变换)就归结为计算两个 $(N/2-1)$ 次多项式 $p_e(t)$ 和 $p_o(t)$ 在 $\omega_{N/2}^0,\omega_{N/2}^1,\omega_{N/2}^2,\cdots,\omega_{N/2}^{N/2-1}$ 这 $N/2$ 个点上的值(即两个 $N/2$ 个数据构成的序列的离散傅里叶变换)。如果 $N=2^m$(m 为正整数)，那么这样的归结过程可以不断进行下去，使计算量大大减少，这就是快速傅里叶变换的基本思路。式(2.111)～式(2.117)的主要计算过程如下：

(1) 根据已知离散数据 $a(j)=a_j(j=0,1,2,\cdots,N-1)$，给多项式 $p_e(t)$ 和 $p_o(t)$ 的系数赋值：

$$a_e(r)=a(2r),\quad a_o(r)=a(2r+1)\quad(r=0,1,2,\cdots,N/2-1) \tag{2.118}$$

(2) 计算

$$v(k)=p_e(\omega_{N/2}^k),\quad u(k)=p_o(\omega_{N/2}^k)\quad(k=0,1,2,\cdots,N/2-1)$$

(3) 最后计算

$$c_k=c(k)=v(k)+\omega_N^k u(k),\quad c_{k+N/2}=c(k+N/2)=v(k)-\omega_N^k u(k)\quad(k=0,1,2,\cdots,N/2-1)$$

程序(2.8) 根据公式(2.110)～公式(2.117)计算离散傅里叶变换的MATLAB程序。

程序任务：用公式(2.110)～公式(2.117)完成对给定数据序列的离散傅里叶变换。

```
function c = FourierTran (a)
% 输入:a—— 离散数据向量(1 × N),N 应当是偶数
```

```
% 输出:c—— 输入数据的离散傅里叶变换的结果向量(1 × N)
N = length(a);                                   % 提取输入数据数目
ae = zeros(1, N/2); ao = zeros(1, N/2);          % 式(2.112) 和式(2.113) 中的系数向量
for j = 1:1:(N/2)
      ae(j) = a(2 * j - 1); ao(j) = a(2 * j);    % 公式(2.118)
end
w1 = exp(- 2 * pi * i/(N/2));                    %i = sqrt(- 1)
v = zeros(1, N/2); u = zeros(1, N/2);            % 式(2.116) 中的 v 和 u 向量
for m = 1:1:(N/2)
      v(m) = 0; u(m) = 0;
      for k = (N/2): - 1:1
            v(m) = ae(k) + v(m)  * w1^(m - 1);   % 公式(2.117)
            u(m) = ao(k) + u(m) *  w1^(m - 1);   % 公式(2.117)
      end
end
w2 = exp(- 2 * pi * i/N);
c = zero(1,N);                                   % 式(2.110) 中的 c 向量
for n = 0:1:(N/2)
      c(n) = v(n) + (w2^(n - 1)) * u(n);         % 公式(2.116)
      c(n + N/2) = v(n) - (w2^(n - 1)) * u(n);   % 公式(2.116)
end
```

例 2.8　应用程序(2.8) 完成对数据组{1,2,3,4,5,6,7,8} 的傅里叶变换。

在 MATLAB 命令窗口中输入:

```
>> a=[1,2,3,4,5,6,7,8];          % 输入已知数据向量
>> c=FourierTran(a);             % 调用程序(2.8) 完成计算
>> c'                            % 将结果以列向量形式输出
```

输出结果:

```
ans =
      36.0000
      - 4.0000 - 9.6569i
      - 4.0000 - 4.0000i
      - 4.0000 - 1.6569i
      - 4.0000
      - 4.0000 + 1.6569i
      - 4.0000 + 4.0000i
      - 4.0000 + 9.6569i
```

下面以 $N=2^3=8$ 为例,详细介绍快速傅里叶变换的算法。此时,离散傅里叶变换式(2.97) 是

$$c_k=\sum_{j=0}^{7}a_j\omega^{kj}\quad(k=0,1,2,\cdots,7) \tag{2.119}$$

设正整数 n 除以 N 的余数是 r,由于 $\omega^N=1$,则有 $\omega^n=\omega^r$。将 k 和 j 用二进制表示为

$$k=k_2 2^2+k_1 2^1+k_0 2^0 \triangleq (k_2,k_1,k_0),\quad j=j_2 2^2+j_1 2^1+j_0 2^0 \triangleq (j_2,j_1,j_0) \tag{2.120}$$

其中,k_s 与 $j_s(s=0,1,2)$ 取 0 或 1。同时,引入记号

$$c_k=c(k_2,k_1,k_0),\quad a_j=a(j_2,j_1,j_0)$$

则式(2.119) 改写为

$$\begin{aligned}c(k_2,k_1,k_0)&=\sum_{j_0=0}^{1}\sum_{j_1=0}^{1}\sum_{j_2=0}^{1}a(j_2,j_1,j_0)\,\omega^{(k_2,k_1,k_0)\,(j_2,j_1,j_0)}=\\&\sum_{j_0=0}^{1}\left\{\sum_{j_1=0}^{1}\left[\sum_{j_2=0}^{1}a(j_2,j_1,j_0)\,\omega^{k_0(j_2,j_1,j_0)}\right]\omega^{k_1(j_1,j_0,0)}\right\}\omega^{k_2(j_0,0,0)}\end{aligned} \tag{2.121}$$

再次引入记号

$$\left.\begin{aligned}&A_0(j_2,j_1,j_0)=a(j_2,j_1,j_0)\\&A_1(j_1,j_0,k_0)=\sum_{j_2=0}^{1}A_0(j_2,j_1,j_0)\,\omega^{k_0(j_2,j_1,j_0)}\\&A_2(j_0,k_1,k_0)=\sum_{j_1=0}^{1}[A_1(j_1,j_0,k_0)]\,\omega^{k_1(j_1,j_0,0)}\\&A_3(k_2,k_1,k_0)=\sum_{j_0=0}^{1}A_2(j_0,k_1,k_0)\,\omega^{k_2(j_0,0,0)}\end{aligned}\right\} \tag{2.122}$$

式(2.121) 就成为

$$c(k_2,k_1,k_0)=A_3(k_2,k_1,k_0)$$

这样就将 c_k 的计算分成了三步。注意到,当 $N=2^m$ 时,$\omega^{k_s 2^{m-1}}=(\omega^{N/2})^{k_s}=(-1)^{k_s}$,公式(2.122) 中第二式还能进一步化简为

$$\begin{aligned}A_1(j_1,j_0,k_0)=&\sum_{j_2=0}^{1}A_0(j_2,j_1,j_0)\,\omega^{k_0(j_2,j_1,j_0)}=A_0(0,j_1,j_0)\,\omega^{k_0(0,j_1,j_0)}+A_0(1,j_1,j_0)\,\omega^{k_0(1,j_1,j_0)}=\\&A_0(0,j_1,j_0)\,\omega^{k_0(0,j_1,j_0)}+A_0(1,j_1,j_0)\,\omega^{k_0 2^2}\omega^{k_0(0,j_1,j_0)}=\\&[A_0(0,j_1,j_0)+(-1)^{k_0}A_0(1,j_1,j_0)]\,\omega^{k_0(0,j_1,j_0)}\end{aligned}$$

则

$$\left.\begin{aligned}&A_1(j_1,j_0,0)=A_0(0,j_1,j_0)+A_0(1,j_1,j_0)\\&A_1(j_1,j_0,1)=[A_0(0,j_1,j_0)-A_0(1,j_1,j_0)]\,\omega^{(0,j_1,j_0)}\end{aligned}\right\} \tag{2.123}$$

应用式(2.120),把括号内的标号还原为十进制,并令

$$j=(0,j_1,j_0)=j_1 2^1+j_0 2^0\quad (j=0,1,2,3)$$

就有

$$\begin{aligned}&2j=j_1 2^2+j_0 2^1=(j_1,j_0,0)\\&2j+1=j_1 2^2+j_0 2^1+2^0=(j_1,j_0,1)\\&j+2^2=2^2+j_1 2^1+j_0 2^0=(1,j_1,j_0)\end{aligned}$$

式(2.123) 改写为

$$\left.\begin{aligned}&A_1(2j)=A_0(j)+A_0(j+2^2)\\&A_1(2j+1)=[A_0(j)-A_0(j+2^2)]\,\omega^j\end{aligned}\quad (j=0,1,2,3)\right\} \tag{2.124}$$

用同样的方法可推导出

$$\left.\begin{aligned}A_2(j2^2+k)&=A_1(2j+k)+A_1(2j+k+2^2)\\A_2(j2^2+k+2)&=[A_1(2j+k)-A_1(2j+k+2^2)]\omega^{2j}\end{aligned}\quad(j,k=0,1)\right\}\tag{2.125}$$

与

$$\left.\begin{aligned}A_3(k)&=A_2(k)+A_2(k+2^2)\\A_3(k+2^2)&=A_2(k)-A_2(k+2^2)\end{aligned}\quad(k=0,1,2,3)\right\}\tag{2.126}$$

根据公式(2.124) ~ 公式(2.126)，由 $A_0(j)=a(j)=a_j(j=0,1,\cdots,7)$ 逐步计算到 $A_3(k)\ (k=0,1,\cdots,7)$，就得到了 $c(k)=c_k$，完成了傅里叶变换的计算。

对于一般 $N=2^m$ 的情况，快速傅里叶变换的算法如下：

$$\left.\begin{aligned}A_l(j2^l+k)&=A_{l-1}(j2^{l-1}+k)+A_{l-1}(j2^{l-1}+k+2^{m-1})\\A_l(j2^l+k+2^{l-1})&=[A_{l-1}(j2^{l-1}+k)-A_{l-1}(j2^{l-1}+k+2^{m-1})]\omega^{j2^{l-1}}\\&(l=1,2,\cdots,m;j,k=0,1,\cdots,2^{m-l}-1)\end{aligned}\right\}\tag{2.127}$$

计算有二重循环，第一重循环是 $l=1,2,\cdots,m$，第二重循环是 $j,k=0,1,\cdots,2^{m-l}-1$。这就是快速傅里叶变换的算法，其运算量是 $\frac{3}{2}N\log_2 N$。当 N 比较大时，快速傅里叶变换的运算次数比矩阵乘法运算次数少很多，如 $N=64,256,1\ 024$ 时，两种方法运算次数之比分别是 0.070 9，0.023 5，0.007 3，大幅度减少计算机运行时间。

程序(2.9)　快速傅里叶变换的 MATLAB 程序。

程序任务：根据公式(2.127)完成对给定数据的快速傅里叶变换。

```
function c = FastFouTran(a)
% 输入:a—— 离散数据向量(1 × N),N 应当等于 2^m
% 输出:c—— 输入数据傅里叶变换的结果向量(1 × N)
N = length(a);   % 提取输入数据数量
m = log2(N);     % 计算 m
if fix(m) ~= m   % 保证 N = 2^m
    error('N 必须是 2 的整数次幂 ');
    return
end
i = sqrt(-1);    %i 是虚数单位
A1 = zeros(1, N); A2 = zeros(1, N); w = zeros(1, N/2);    % 初始化有关矩阵
A1(:) = a(:);                                              % 赋傅里叶变换的数据
w0 = exp(-2 * pi * i/N);                     % 定义 w = exp(-2πi/N)。逆变换时 w = exp(2πi/N)
w(1:N/2) = w0.^(0:(N/2-1));                                % 建立 w^j(j = 0,1,…,N/2-1) 向量
for l = 1:m
    if rem(l, 2) ~= 0                                      %l 为奇数时进行下列运算
        for k = 0:(2^(m-l)-1)
            for j = 0:(2^(l-1)-1)
                A2(k * 2^l+j+1) = A1(k * 2^(l-1)+j+1) + ...
                    A1(k * 2^(l-1)+j+2^(m-1)+1);       % 公式(2.127)
                A2(k * 2^l+j+2^(l-1)+1) = (A1(k * 2^(l-1)+j+1) - ...
                    A1(k * 2^(l-1)+j+2^(m-1)+1)) * w((k * 2^(l-1))+1);
```

```
            end
        end
    else                                                        %l 为偶数时进行下列运算
        for k = 0:(2^(m - l) - 1)
            for j = 0:(2^(l - 1) - 1)
                A1(k * 2^l + j + 1) = A2(k * 2^(l - 1) + j + 1) + …
                    A2(k * 2^(l - 1) + j + 2^(m - 1) + 1);      % 公式(2.127)
                A1(k * 2^l + j + 2^(l - 1) + 1) = (A2(k * 2^(l - 1) + j + 1) - …
                    A2(k * 2^(l - 1) + j + 2^(m - 1) + 1)) * w((k * 2^(l - 1)) + 1);
            end
        end
    end
end
if rem(m, 2) == 0
    A2(:) = A1(:);
end
c = A2;      % 输出快速傅里叶变换结果。逆变换时 c = A2/N
```

例 2.9 一个应用快速傅里叶变换进行频谱分析的简单例子。

(1) 建立一个离散时间序列：

```
>> t=0:0.001:0.001 * (2^8 - 1);
```

(2) 构造包含两种频率正弦信号的离散数据组：

```
>> x =2 * sin(2 * pi * 40 * t) + 4 * sin(2 * pi * 90 * t);
```

画出 x 的分布曲线：

```
>> plot(x(1:120));
```

显然，数据分布有一定的规律性(见图 2.11)。

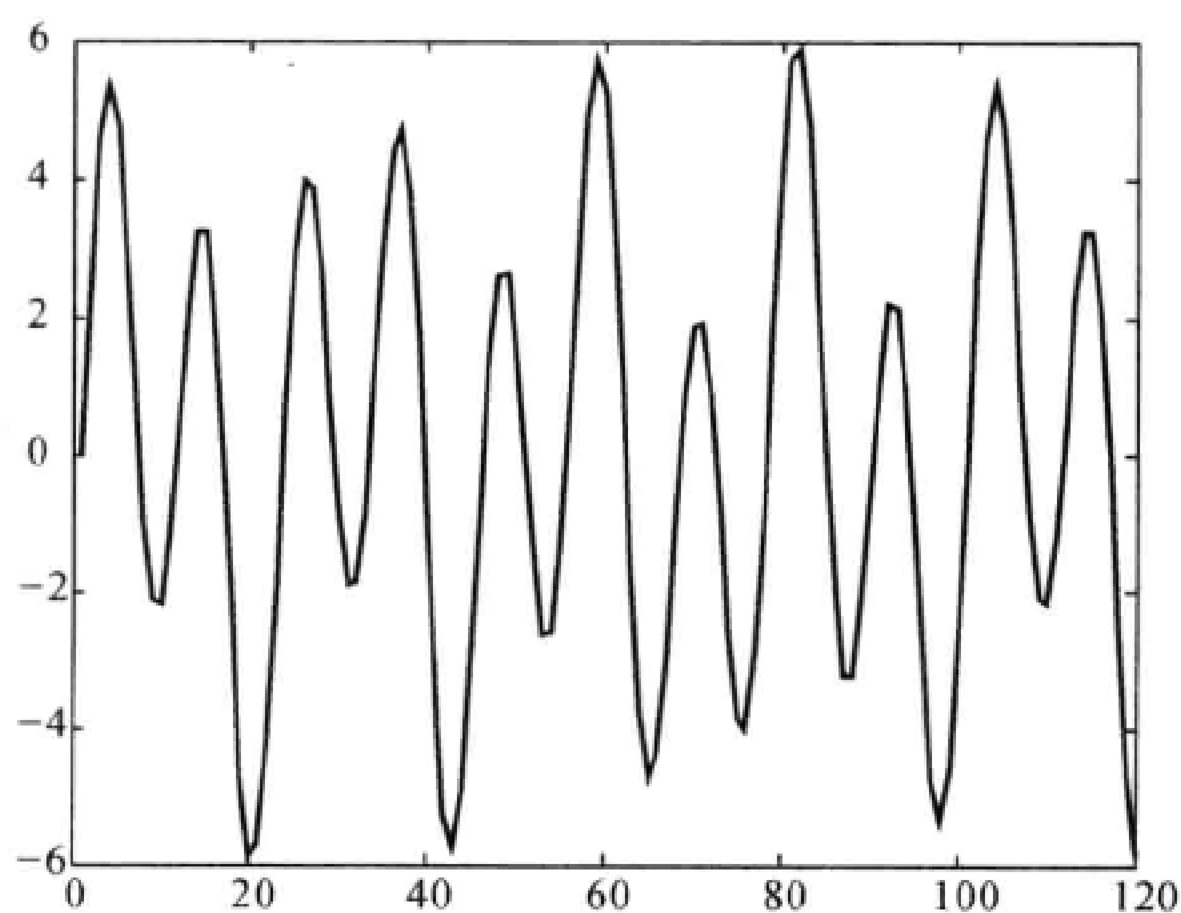


图 2.11

(3) 给 x 叠加一个噪声:

```
>> y=x+3*randn(size(t));
```

画出 y 的分布曲线:

```
>> plot(y(1:120));
```

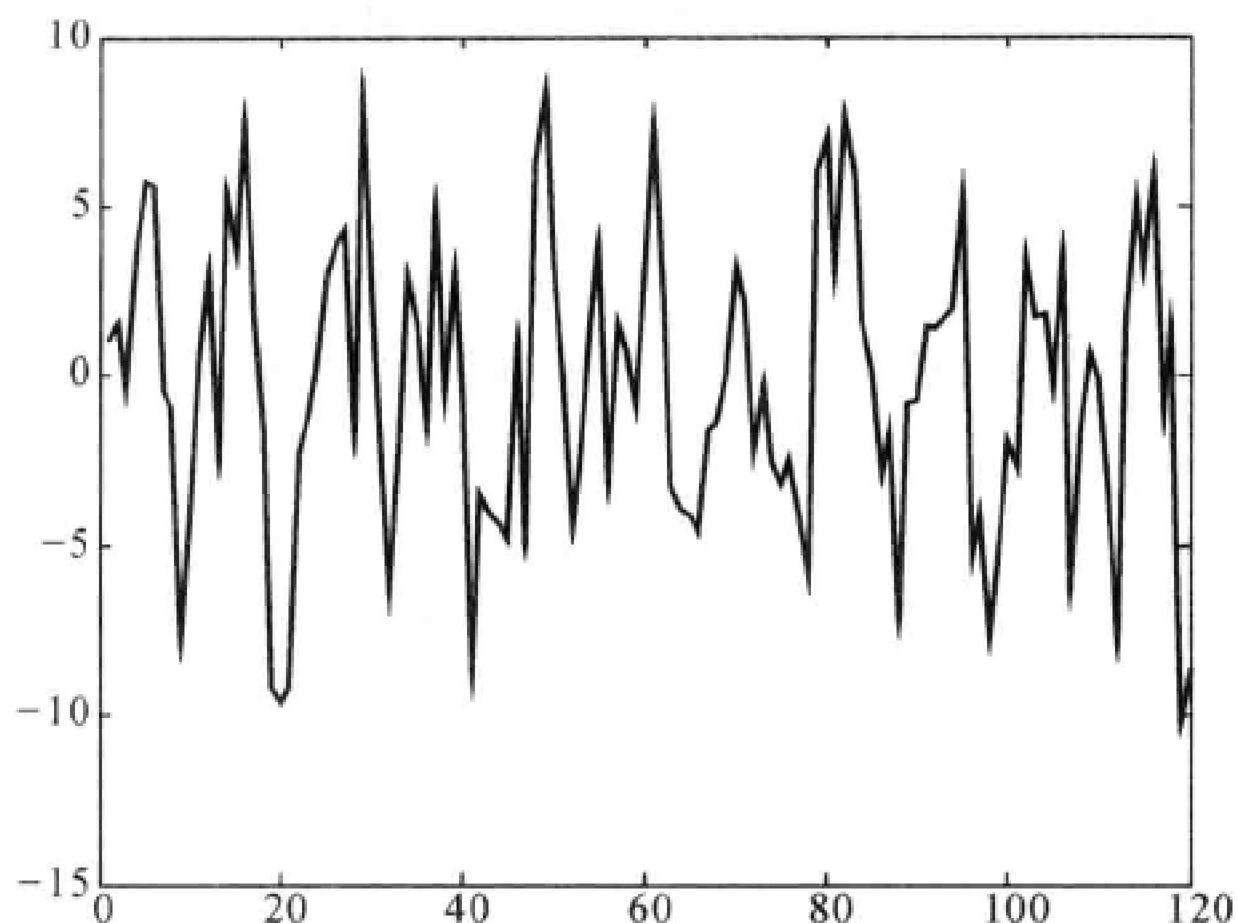


图　2.12

从图 2.12 中很难看出数据分布的规律性和所包含的频率成分。

(4) 对数据 y 进行傅里叶变换,并绘制其功率谱密度曲线:

```
>> c=FastFouTran(a);                              % 应用程序(2.9)完成傅里叶变换
>> p=c.*conj(c);                                  % 计算功率谱密度
>> plot(1000/256*(0:99),p(1:100)/1000)            % 绘制功率谱密度曲线
```

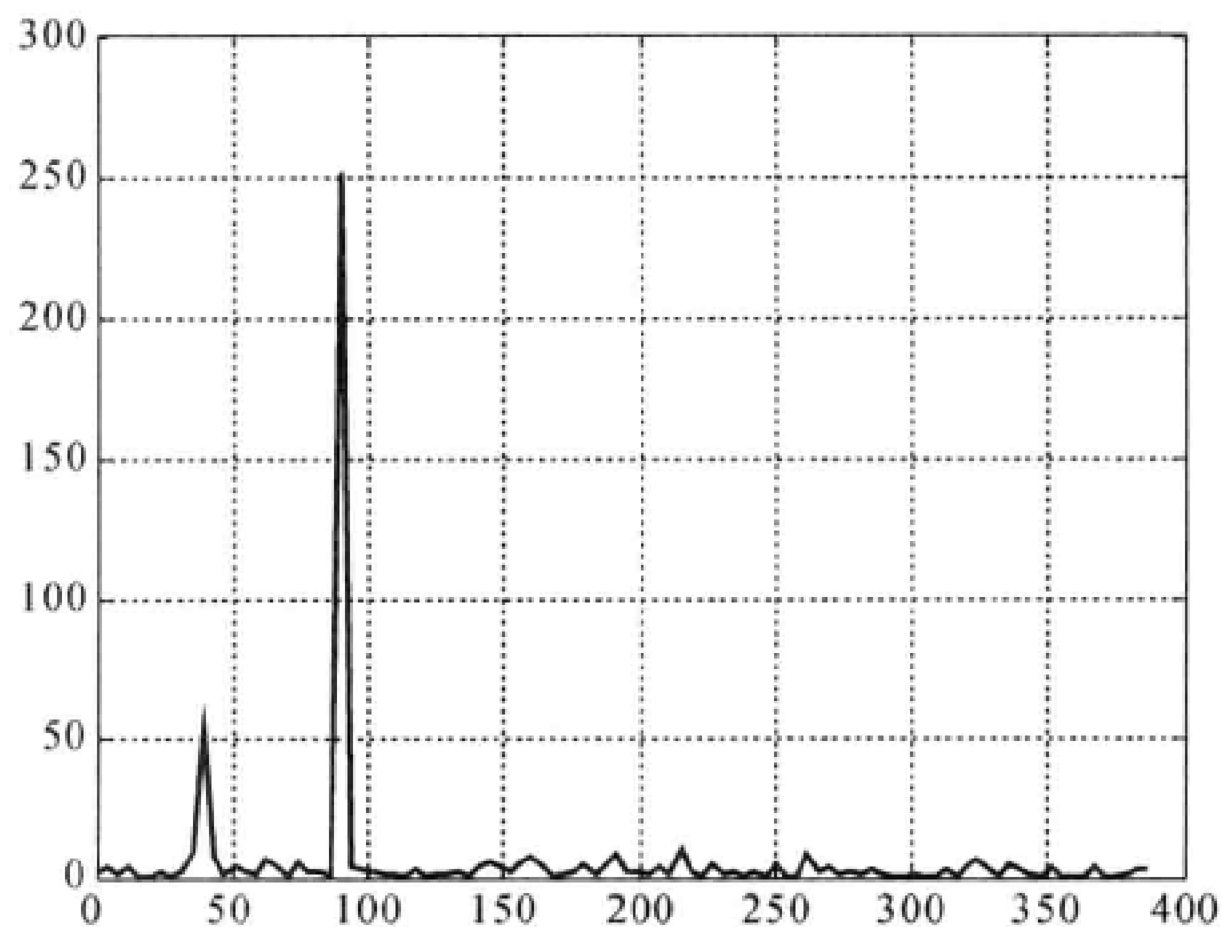


图　2.13

功率谱密度曲线在 40 和 90 处有两个尖峰,这正是数据中包含的两种频率(见图 2.13)。

2.7 物理学中的应用举例

2.7.1 大角度振动单摆的周期

鞠衍清[①]用线性插值法研究了单摆运动周期。这里对其主要思路进行简单介绍。

在忽略阻力的情况下，单摆运动的动力学方程为

$$ml\ddot{\theta} = -mg\sin\theta \tag{2.128}$$

式中，m 为摆球质量，l 为摆长，g 为重力加速度，θ 为摆角。在小摆角的情况下，可以利用 $\sin(\theta) \approx \theta$ 的近似，将单摆的运动作为简谐振动来研究，得到振动周期是

$$T_0 = 2\pi\sqrt{\frac{l}{g}} \tag{2.129}$$

但当摆角较大(一般认为大于 5°)时，该近似便不适用了。此时单摆的振动周期必须用方程式(2.128)的精确解求出，即

$$T = \frac{2T_0}{\pi}\int_0^{\frac{\pi}{2}} \frac{\mathrm{d}\theta}{\sqrt{1-\sin^2(\theta_0/2)\sin^2\theta}} \tag{2.130}$$

式中，θ_0 为摆角 θ 的幅值。上式涉及第一类椭圆积分，由于被积函数分母难以处理，因而积分不易求解。

如果能找到一个在$[0,\pi/2]$范围内与式(2.130)中被积函数分母比较接近的简单函数，则积分便可以方便地求出。为了方便计算，令 $k=\sin(\theta_0/2)$，$f(\theta,k)=\sqrt{1-k^2\sin^2\theta}$，则式(2.130)改写为

$$T = \frac{2T_0}{\pi}\int_0^{\frac{\pi}{2}} \frac{\mathrm{d}\theta}{f(\theta,k)} \tag{2.131}$$

通过绘制 $f(\theta,k)$-θ 曲线发现，对于$[0,\pi/2]$范围内的不同 θ_0(即不同 k)，函数 $f(\theta,k)$ 是一个波浪状的平滑函数，并且随着 θ_0 的减小，其"波浪"的起伏也随之减小。根据这一特性，可以用插值法，针对不同的 k，用一条直线来替代曲线 $f(\theta,k)$。该直线通过点(0,1)与点$(\pi/2,\sqrt{1-k^2})$，方程是

$$r(\theta,\theta_0) = 1 - \frac{2}{\pi}(1-a)\theta \tag{2.132}$$

式中，$a=\sqrt{1-k^2}=\cos(\theta_0/2)$。将式(2.132)代入式(2.131)进行积分，得到单摆振动周期的一个近似公式

$$T = T_0\frac{\ln a}{a-1} = T_0\frac{\ln\cos\dfrac{\theta_0}{2}}{\cos\dfrac{\theta_0}{2}-1} \tag{2.133}$$

数值计算结果的比较表明，式(2.133)是精确解式(2.130)的一个很好的近似。

---

① 鞠衍清. 用线性插值法求单摆运动周期的近似解. 大学物理，2006，25(12)：32－34.

2.7.2　用单缝衍射方法测量波长

花世群[①]把最小二乘函数拟合方法应用于单缝衍射测量光波波长实验数据的处理，显著提高了测量结果的准确度。

单缝衍射图样的光强分布为

$$I(x)=I_0\left(\frac{\sin u}{u}\right)^2 \tag{2.134}$$

式中，$u=\pi xd/D\lambda$，I_0 为中央明纹中心的强度，λ 为光波波长，d 为衍射狭缝宽度，D 为狭缝到观察屏的距离，x 为观察屏上的位置坐标。设 x_{01} 和 x_{02} 分别为衍射图样中央明纹两侧一级暗纹中心的位置坐标，则中央明纹宽度 $\Delta x_0=x_{02}-x_{01}$，并且

$$\lambda=\frac{\Delta x_0 d}{2D} \tag{2.135}$$

由于测量系统 $D=2\,000\text{mm}$，误差 $|\Delta D|\leqslant 2\text{mm}$，对于给定狭缝，根据上式来测量波长的关键是如何提高中央明纹宽度 Δx_0（亦即 x_{01} 和 x_{02}）的测量精度，难点在于准确定位一级暗纹中心的位置，为此采用最小二乘拟合方法处理测量数据。

把式(2.134)中的正弦函数用泰勒级数展开，就有

$$I(x)=I_0\left[1-\frac{1}{3}\left(\frac{\pi d}{D\lambda}\right)^2 x^2+\frac{2}{45}\left(\frac{\pi d}{D\lambda}\right)^4 x^4-\cdots\right] \tag{2.136}$$

由上式可知，衍射图样的强度分布曲线在衍射条纹中的某一小段（如暗条纹中心附近）内与抛物线形状相似，因此可以用二次多项式

$$f(x)=a_0+a_1x+a_2x^2 \tag{2.137}$$

对式(2.136)所表示的同一段衍射条纹强度分布进行拟合。

设实验在等精度情况下测量，观察屏上第一级衍射暗条纹中心附近强度分布的测量数据为 (x_i,I_i) $(i=1,2,\cdots,N)$，则利用最小二乘法原理，用多项式(2.137)对衍射暗条纹中心附近的强度分布进行拟合，得出多项式(2.137)中的三个系数 a_0，a_1 和 a_2。若观察屏上暗条纹中心的位置为 x_0，则有 $[\mathrm{d}f(x)/\mathrm{d}x]_{x=x_0}=0$，由式(2.137)得到

$$x_0=-\frac{a_1}{2a_2} \tag{2.138}$$

测量数据见表 2.13。

表　2.13

左		右	
x_i/mm	I_i/格	x_i/mm	I_i/格
17.40	75	30.00	126
17.60	37	30.20	69
17.80	16	30.40	36
18.00	14	30.60	21

① 花世群. 最小二乘法在单缝衍射测波长中的应用. 大学物理，2005，24(2)：40－42.

续 表

左		右	
x_i/mm	I_i/格	x_i/mm	I_i/格
18.20	33	30.80	25
18.40	73	31.00	42
18.60	140	31.20	70

用曲线拟合，得出中央明纹左侧和右侧一级衍射暗条纹中心所在处的光强度分布分别是

$$f_1(x)=83\ 187-9\ 292.1x+259.52x^2$$

$$f_2(x)=211\ 008-13\ 752.7x+224.11x^2$$

再根据式(2.138)分别计算一级暗条纹中心坐标 x_{01} 和 x_{02} 及中央明纹宽度 Δx_0，代入式(2.135)计算波长测量值 λ，与理论值 $\lambda_0=6.328\times10^{-4}$ mm 相比较计算百分差 E_0。数据处理结果见表 2.14(狭缝宽度 $d=0.918$mm)。

表 2.14

数据处理方法	x_{01}/mm	x_{02}/mm	Δx_0/mm	$\lambda/(10^{-4}$ mm)	E_0/(%)
非最小二乘法	18.1	30.6	12.6	6.2±0.2	2.0
最小二乘法	17.9	30.7	12.8	6.3±0.1	0.4

两种数据处理结果的比较表明，测量结果的百分差相差一个数量级，说明用最小二乘法对衍射暗条纹进行曲线拟合，能够精确定位暗条纹中心的位置，显著提高测量精度。

2.7.3 金属电阻温度系数的测量

物质的电阻随温度变化。对于金属，温度在一定范围内升高时电阻增大。电阻与温度的关系通常用下列经验公式表示：

$$R_t=R_0(1+\alpha t+\beta t^2+\gamma t^3+\cdots) \tag{2.139}$$

式中，R_t 和 R_0 分别是温度为 t(℃)和 0℃ 时的电阻值，α,β,γ 分别是一、二、三级电阻温度系数。纯金属在温度不太高时，电阻和温度的关系近似为线性，即

$$R_t=R_0(1+\alpha t) \tag{2.140}$$

只要测量出不同温度时的电阻值，用作图法或最小二乘直线拟合法求出直线的截距 R_0 和斜率 $K=R_0\alpha$，就可以计算出电阻的温度系数

$$\alpha=\frac{K}{R_0} \tag{2.141}$$

某次实验的测量数据见表 2.15。

表 2.15

t/℃	83.9	74.8	66.5	57.9	48.6	40.1	33.8	26.5
R_t/Ω	0.467 1	0.452 3	0.440 3	0.428 3	0.415 5	0.409 5	0.393 4	0.382 5

程序(2.10)　计算电阻温度系数的 MATLAB 程序。

程序任务：根据温度和电阻的测量数据(t_i,R_i) $(i=1,2,\cdots,8)$，用最小二乘法拟合直线$R=Kt+R_0$，并计算电阻温度系数α。

```
x = [83.9, 74.8, 66.5, 57.9, 48.6, 40.1, 33.8, 26.5];
y = [0.4671, 0.4523, 0.4403, 0.4283, 0.4155, 0.4095, 0.3934, 0.3825];
xm = mean(x);
ym = mean(y);
sx = (x - xm) * (x - xm)';
sy = (y - ym) * (x - xm)';
K = sy/sx;
R0 = ym - K. * xm
alfa = K/R0
```

运行程序的输出结果如下：

```
R0 =0.3468
alfa =0.004103
```

即 $R_0=0.346\ 8\Omega$，$\alpha=4.103\times10^{-3}/℃$。

2.7.4　菲涅尔衍射向夫琅禾费衍射的过渡

在如图 2.14 所示的单色光衍射系统中，$\xi O\eta$ 平面是衍射屏所在平面，xoy 平面是观察屏所在平面，两个平面相互平行。直角坐标系 $O\xi\eta$ 和 oxy 对应坐标轴相互平行，两坐标原点连线与两坐标平面垂直，长度为 z。一束单色光从衍射屏的左边照射衍射屏，透过衍射屏的光场复振幅为 $U'(\xi,\eta)$，根据瑞利-索末菲衍射公式，观察屏上的光场复振幅是

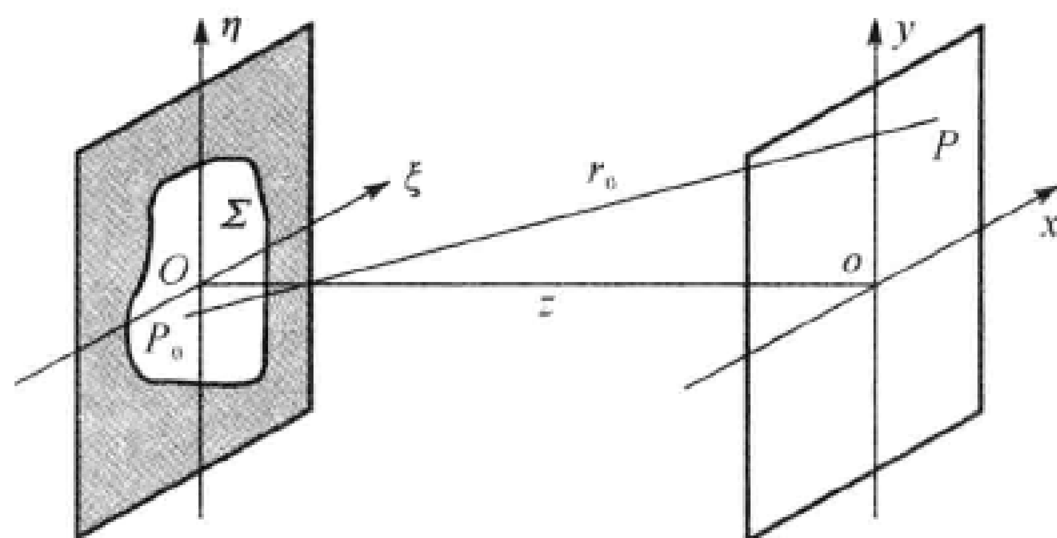


图　2.14

$$U(x,y)=\frac{z}{\mathrm{j}\lambda}\iint_{\Sigma}U'(\xi,\eta)\frac{\exp(\mathrm{j}kr_0)}{r_0^2}\mathrm{d}\xi\mathrm{d}\eta \tag{2.142}$$

式中，$r_0=\sqrt{z^2+(x-\xi)^2+(y-\eta)^2}$，$\mathrm{j}=\sqrt{-1}$，$k=2\pi/\lambda$，$\lambda$ 是光波波长。在菲涅耳近似条件满足时，即 $z^3\gg\frac{\pi}{4\lambda}[(x-\xi)^2+(y-\eta)^2]^2_{\max}$，式(2.142) 简化为菲涅耳衍射公式

$$U(x,y)=\frac{\exp(\mathrm{j}kz)}{\mathrm{j}\lambda z}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}U'(\xi,\eta)\exp\left\{\frac{\mathrm{j}k}{2z}[(x-\xi)^2+(y-\eta)^2]\right\}\mathrm{d}\xi\mathrm{d}\eta \tag{2.143}$$

如果还满足更强的夫琅禾费近似，即 $z\gg k(\xi^2+\eta^2)_{\max}/2$，式(2.143) 又简化为夫琅禾费衍射

公式

$$U(x,y)=\frac{\exp(\mathrm{j}kz)\exp\left[\frac{\mathrm{j}k}{2z}(x^2+y^2)\right]}{\mathrm{j}\lambda z}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}U'(\xi,\eta)\exp\left[-\mathrm{j}\frac{2\pi}{\lambda z}(x\xi+y\eta)\right]\mathrm{d}\xi\mathrm{d}\eta \tag{2.144}$$

观察屏上的光强度分布为

$$I(x,y)=|U(x,y)|^2 \tag{2.145}$$

若单位振幅的平行光垂直照射衍射屏，并且衍射屏的复振幅透过率是

$$t(\xi,\eta)=t(\xi) \tag{2.146}$$

则透过衍射屏的光波复振幅为 $U'(\xi,\eta)=t(\xi)$。此时菲涅耳衍射公式(2.143)简化为

$$U(x)=\frac{\exp(\mathrm{j}(kz+\pi/4))}{\mathrm{j}\sqrt{\lambda z}}\exp\left(\frac{\mathrm{j}kx^2}{2z}\right)\int_{-\infty}^{\infty}t(\xi)\exp\left(\frac{\mathrm{j}k\xi^2}{2z}\right)\exp\left[-2\pi\mathrm{j}\left(\frac{x}{\lambda z}\right)\xi\right]\mathrm{d}\xi=$$

$$\frac{\exp(\mathrm{j}(kz+\pi/4))}{\mathrm{j}\sqrt{\lambda z}}\exp\left(\frac{\mathrm{j}kx^2}{2z}\right)F\left\{t(\xi)\exp\left(\frac{\mathrm{j}k\xi^2}{2z}\right)\right\}\Bigg|_{f=x/(\lambda z)} \tag{2.147}$$

其中，$F\{g(\xi)\}\Big|_{f=x/(\lambda z)}$ 表示函数 $g(\xi)$ 的傅里叶变换函数 $G(f)$ 在 $f=x/(\lambda z)$ 处的值。夫琅禾费衍射公式(2.144)简化为

$$U(x,y)=-\mathrm{j}\delta(y)\exp(\mathrm{j}kz)\exp\left(\frac{\mathrm{j}kx^2}{2z}\right)\int_{-\infty}^{\infty}t(\xi)\exp\left[-2\pi\mathrm{j}\left(\frac{x}{\lambda z}\right)\xi\right]\mathrm{d}\xi=$$

$$-\mathrm{j}\delta(y)\exp(\mathrm{j}kz)\exp\left(\frac{\mathrm{j}kx^2}{2z}\right)F\{t(\xi)\}\Big|_{f=x/(\lambda z)} \tag{2.148}$$

如果衍射屏是宽度为 w 的狭缝，即

$$t(\xi)=\mathrm{rect}\left(\frac{\xi}{w}\right)=\begin{cases}1 & (|\xi|\leqslant w/2)\\ 0 & (|\xi|>w/2)\end{cases} \tag{2.149}$$

则菲涅尔衍射的复振幅分布为

$$U(x)=\frac{\exp(\mathrm{j}(kz+\pi/4))}{\mathrm{j}\sqrt{\lambda z}}\exp\left(\frac{\mathrm{j}kx^2}{2z}\right)\int_{-\infty}^{\infty}\mathrm{rect}\left(\frac{\xi}{w}\right)\exp\left(\frac{\mathrm{j}k\xi^2}{2z}\right)\exp\left[-2\pi\mathrm{j}\left(\frac{x}{\lambda z}\right)\xi\right]\mathrm{d}\xi=$$

$$\frac{w\exp(\mathrm{j}(kz+\pi/4))}{\mathrm{j}\sqrt{\lambda z}}\exp\left(\frac{\mathrm{j}kx^2}{2z}\right)F\left\{\mathrm{rect}(\xi)\exp\left(\mathrm{j}\frac{\pi w^2}{\lambda z}\xi^2\right)\right\}\Bigg|_{f=wx/(\lambda z)} \tag{2.150}$$

强度分布为

$$I(x)=\frac{w^2}{\lambda z}\left|F\{\mathrm{rect}(\xi)\exp(\mathrm{j}\pi s\xi^2)\}\right|^2_{f=wx/(\lambda z)} \tag{2.151}$$

其中

$$s=\frac{w^2}{\lambda z} \tag{2.152}$$

观察屏上的强度分布与 $\mathrm{rect}(\xi)\exp(\mathrm{j}\pi s\xi^2)$ 的傅里叶变换有关。对于夫琅禾费衍射，观察屏上强度与坐标 x 的关系是

$$I(x)=w^2\left|F\{\mathrm{rect}(\xi)\}^2\right|_{f=xw/(\lambda z)} \tag{2.153}$$

式(2.151)和式(2.153)中傅里叶变换的频率 f 与观察屏上坐标 x 的关系是

$$\frac{x}{w}=\frac{f}{s} \tag{2.154}$$

随着参数 s 的减小(对于确定的 w 和 λ,对应于 z 的增大),式(2.151)描述的菲涅尔衍射的强度分布会平稳过渡到式(2.153)描述的夫琅禾费衍射强度分布,亦即衍射从菲涅尔衍射区过渡到夫琅禾费衍射区。

假设 $w=1$,取不同的 s 值,应用快速傅里叶变换算法和程序,完成式(2.151)中的傅里叶变换,并计算观察屏上的归一化强度分布,分析菲涅尔衍射区与夫琅禾费衍射区之间的过渡区所对应的参数 s 的近似值。图 2.15 给出了计算结果,图中的纵坐标是观察屏上的归一化强度,横坐标是衍射光的频率 $f=wx/(\lambda z)$。从图中强度分布,可以总结出表 2.16 的数据。

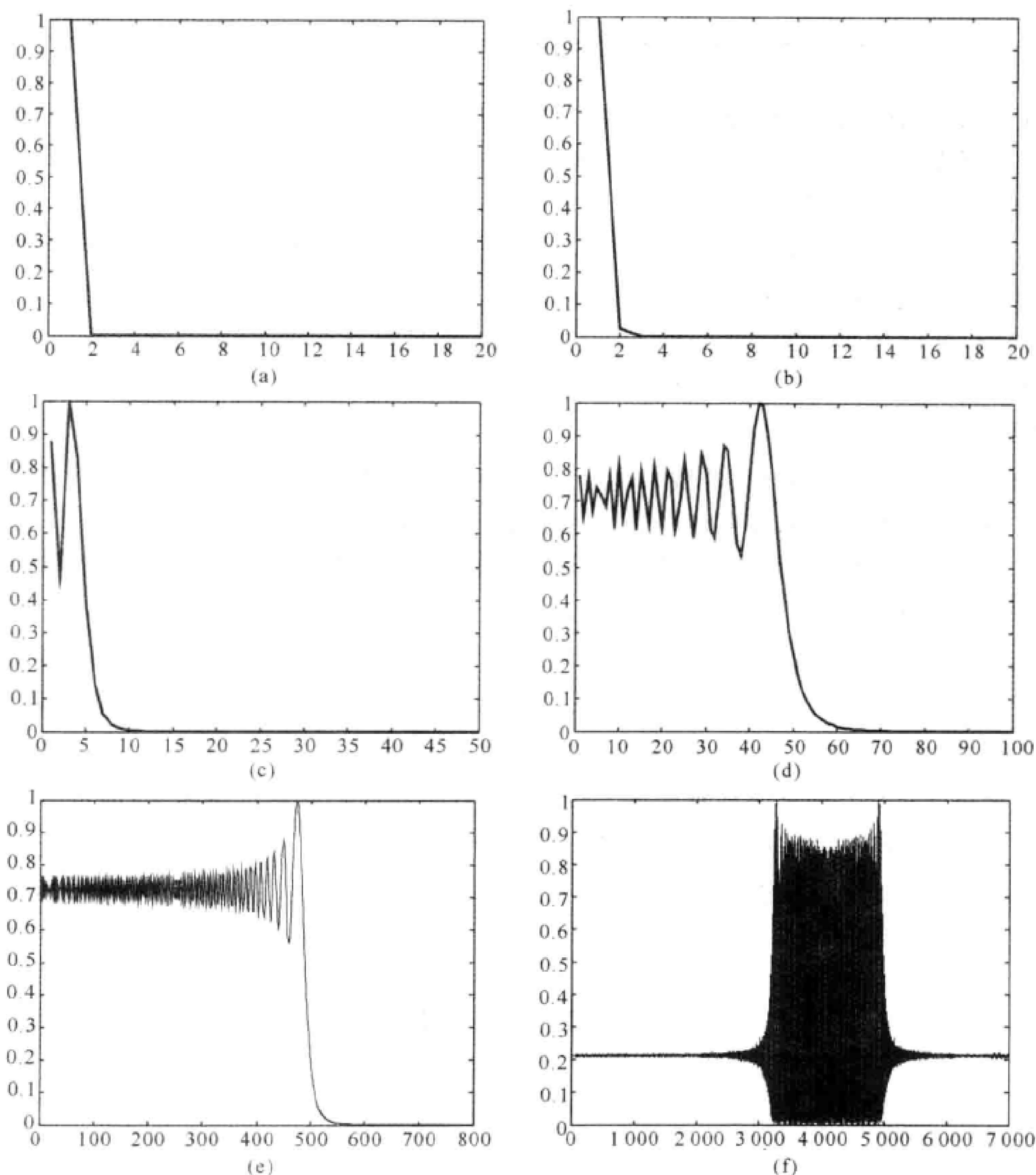


图　2.15

(a) $s=10^{-1}$;　(b) $s=10^{0}$;　(c) $s=10^{1}$

(d) $s=10^{2}$;　(e) $s=10^{3}$;　(f) $s=10^{4}$

表 2.16

图号	s	$\sqrt{\lambda z}/w$	f_m	x_m	f_0	x_0
(a)	10^{-1}	3.16	2	20	2	20
(b)	10^{0}	1.00	2	2.0	3	3.0
(c)	10^{1}	0.316	5	0.5	10	1.0
(d)	10^{2}	0.10	50	0.5	66	0.66
(e)	10^{3}	0.031 6	500	0.5	550	0.55
(f)	10^{4}	0.01	5 000	0.5	6 000	0.6

在表 2.16 中，f_m 是归一化强度有较大值时频率 f 的上限，x_m 是归一化强度有较大值时观察屏上空间坐标 x 的上限，f_0 是归一化强度不等于零处 f 的上限，x_0 是归一化强度不等于零处 x 的上限。分析表中数据，可得以下结论：

(1) 当 $s<10^0$ 时，随着 s 的减小(即 $\sqrt{\lambda z}/w$ 增大，相当于观察屏到衍射屏的距离越来越远)，观察屏上归一化强度有较大值的衍射光频率上限 f_m 基本不变(计算发现，当 $s<10^{-1}$ 时，f_m 确实不变)，说明能够到达衍射屏上的衍射光波频率的最高值保持不变。结合 $f=wx/(\lambda z)$ 可得，此时随着观察屏的远离，观察屏上的衍射图样等比例放大，但强度分布保持不变，这就是夫琅禾费衍射。

(2) 当 $s>10^3$ 时，随着 s 的增大(即 $\sqrt{\lambda z}/w$ 减小，相当于观察屏非常靠近衍射屏并且越来越近)，观察屏上归一化强度有较大值的衍射光频率上限 f_m 急骤增大，并且衍射屏上强度分布在大范围内快速振荡，计算结果不稳定。这说明菲涅耳衍射积分已经不能给出合理结果，应当应用其他衍射理论来分析。

(3) 在 $10^{-1}\sim10^3$ 范围内，随着 s 的增大(即 $\sqrt{\lambda z}/w$ 减少，相当于观察屏到衍射屏的距离越来越近)，观察屏上归一化强度有较大值的衍射光频率上限 f_m 迅速增大，说明能够到达观察屏上的衍射光波中频率较高的分量快速增多。反过来说，随着 $\sqrt{\lambda z}/w$ 的增大(即观察屏到衍射屏的距离越来越远)，能够到达观察屏上的光波中频率较高的分量快速减少，即衍射光中的高频分量随着传播迅速衰减，这就是倏失波。可以认为，$10^3>s>10^1$(对应 $0.031\,6<\sqrt{\lambda z}/w<0.316$)是菲涅尔衍射区，$10^1>s>10^0$(对应 $0.316<\sqrt{\lambda z}/w<1$)是从菲涅尔衍射区向夫琅禾费衍射区的过渡区。

习　　题

1. 给出测量数据(0.00,1.00),(1.00,5.00),(2.00,3.00),(3.00,6.00)的四个三次拉格朗日插值基函数。

2. 设 $f(x)=\sin x$，问插值区间应取多大才能保证线性插值的误差不大于 $\frac{1}{2}\times10^{-6}$。

3. 根据 $\sqrt[3]{27}=3$，$\sqrt[3]{64}=4$，$\sqrt[3]{125}=5$，构造二次拉格朗日插值多项式，并计算 $\sqrt[3]{100}$ 的近似值，估计近似值的误差限并与实际误差比较。

4. 用表 2.17 给出的数据，由四次拉格朗日插值多项式求出 $x=98$ 处函数的近似值，并确定其误差限。

表　2.17

x_i	93.0	96.0	100.0	104.2	108.7
$f(x_i)$	11.38	12.80	14.70	17.07	19.91

5. 设 $x_i(i=0,1,\cdots,n)$ 为互异节点，证明拉格朗日插值基函数 $l_i(x)$ 具有以下性质：

$$(1)\ \sum_{i=0}^{n} l_i(x)\equiv 1;\qquad (2)\ \sum_{i=0}^{n} x_i^k l_i(x)\equiv x^k\qquad (k=0,1,\cdots,n)。$$

6. 试推导等距节点条件下的拉格朗日插值公式及其余项表达式。

7. 给出下列函数表(见表 2.18)。(1) 作出差商表；(2) 写出牛顿插值多项式；(3) 计算 $N_3(1.2)$。

表　2.18

i	0	1	2	3
x_i	-1.00	2.00	3.00	4.00
$f(x_i)$	3.00	5.00	7.00	5.00

8. 已知数据表(见表 2.19)：

表　2.19

x_i	0	1	2	3
$f(x_i)$	1	2	17	64

应用等距节点的牛顿插值公式分别计算 $x=0.5$ 和 $x=2.5$ 时的函数近似值。

9. 给出下列数据(见表 2.20)。对数据进行线性拟合，并计算均方误差。

表　2.20

x_i	-1.00	-0.50	0.00	0.25
$f(x_i)$	0.22	0.80	2.00	2.50

10. 给出下列数据(见表 2.21)：

表　2.21

x_i	4.0	4.2	4.5	4.7	5.1	5.5	5.9	6.3	6.8	7.1
$f(x_i)$	102.56	113.18	130.11	142.05	167.53	195.14	224.87	256.73	299.50	326.72

分别求出函数 $f(x)$ 的形如 $a_0+a_1x+a_2x^2$ 和 ax^b 的最小二乘拟合公式。

11. 设一发射源的发射强度公式为 $I=I_0e^{-at}$，现测得一组数据如下(见表 2.22)：

表 2.22

t_i	0.2	0.3	0.4	0.5	0.6	0.7	0.8
I_i	3.16	2.38	1.75	1.34	1.00	0.74	0.56

试用最小二乘法确定强度公式中的参数 I_0 和 α。

12. 胡克定律指出 $F=kx$，其中 F 是弹簧上的弹性力(单位为 N)，x 是弹簧的伸长量(单位为 m)，k 是弹簧的倔强系数(单位为 N/m)。根据下列两组数据(见表 2.23 和表 2.24)，分别求解弹簧的倔强系数 k 的值。

表 2.23

x_k	0.02	0.04	0.06	0.08	0.10
F_k	3.6	7.3	10.9	14.5	18.2

表 2.24

x_k	0.02	0.04	0.06	0.08	0.10
F_k	5.3	10.6	15.9	21.2	26.4

13. 自由落体的运动方程是 $d=\frac{1}{2}gt^2$，其中 d 表示物体下落距离(单位为 m)，t 表示物体自由下落时间(单位为 s)，g 表示当地的重力加速度(单位为 m/s^2)。根据下面测量数据(见表 2.25)，计算当地的重力加速度。

表 2.25

t_k	0.200	0.400	0.600	0.800	1.000
d_k	0.196 0	0.783 5	1.763 0	3.134 5	4.897 5

14. 一小灯泡上电压和电流的测量值如下(见表 2.26)：

表 2.26

U/V	0.00	0.50	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.50	5.00
I/mA	0.0	27.5	35.2	43.3	51.2	56.1	63.2	70.2	76.3	81.9	86.9

试用最小二乘多项式拟合方法确定均方误差最小、次数最低的 $U-I$ 多项式关系函数。

15. 九大行星到太阳的距离及它们以天为单位的恒星周期如下(见表 2.27)：

表　2.27

行星	距离 /(km × 10^6)	恒星周期 / 天
水星	57.59	87.99
金星	108.11	224.70
地球	149.57	365.26
火星	227.84	686.98
木星	778.14	4 332.4
土星	1 427.0	10 759
天王星	2 870.3	30 684
海王星	4 499.9	60 188
冥王星	5 909.0	90 710

用它们拟合行星第三运动定律的最小二乘幂函数拟合曲线 $y=Cx^{3/2}$。

16. 一单摆的悬线长度 $l=1.500\text{m}$，当它作大幅度摆动时，测得其角振动幅度 θ_m（单位为(°)）与振动周期 T（单位为 s）的数据关系如下（见表 2.28）：

表　2.28

θ_m	1	2	3	4	5	6	7	8	9	10
T	2.458 22	2.458 36	2.458 60	2.458 92	2.459 35	2.459 86	2.460 47	2.461 18	2.461 97	2.462 87

(1) 选用 $\theta_m=1,3,5$ 的数据，应用插值法，建立 T 与 θ_m 的二次多项式关系；

(2) 选用 $\theta_m=1,3,5,7,9$ 的数据，应用最小二乘多项式拟合方法，建立 T 与 θ_m 的二次多项式关系；

(3) 利用全部数据，应用最小二乘多项式拟合方法，建立 T 与 θ_m 的三次多项式关系；

(4) 根据以上三个多项式，分别确定单摆小角度振动时的周期，并据此计算当地的重力加速度。

17. 测得一个连续信号的一组数据 $a=\{4, 3, 2, 1, 0, 1, 2, 3\}$，用 FFT 算法计算其离散频谱 c。

18. 已知方波的一组数据：

$x=\{0.0, 0.1, 0.2, 0.3, \cdots, 2.0\}$，

$y=\{0, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$

用快速傅里叶变换和逆变换对这组数据进行处理，画出图形，分析其频谱特点，并探讨原方波图形与傅里叶逆变换数据图形的关系。

第3章　数值积分与数值微分

微分描述函数的变化，积分反映函数的积累，微分和积分既是物理学的理论基础，也是物理学理论的组成部分，因为许多物理学定律和物理过程（如牛顿第二定律、哈密顿正则方程、热力学第一定律、费马原理、麦克斯韦电磁场方程组、薛定谔方程、爱因斯坦引力场方程、亥姆霍斯方程、扩散过程、波动过程等）的数学表达式本身就是微分方程或积分方程，估计这也是牛顿建立微积分理论的客观原因之一。物理学中许多变化率的计算（如根据位移计算速度、根据速度计算加速度、根据功计算功率、根据电势计算电场强度、根据磁通量计算感应电动势等）都涉及微分运算，而与之相反的计算过程（如路程、功、转动惯量、光程、电场强度、磁感应强度等）自然用到了积分。从数学上看，微分是求解导数函数的过程，积分是求解原函数的过程，用代数方法完成这两个过程有时会非常复杂，甚至根本不可能，更何况计算以数据形式给出的函数的微分与积分。因此，研究函数微分和积分的数值计算方法，不仅有助于物理学理论的发展，也能推动物理学理论的应用。本章介绍数值积分的插值型方法（牛顿-柯特斯方法、复化求积方法）、龙贝格方法和数值微分的差商方法、理查森外推法及插值型方法，最后给出这些方法在物理学中应用的几个例子。

3.1　数值积分概述

在数学上，一般利用牛顿-莱布尼兹（Newton - Leibniz）公式

$$I[f]=\int_a^b f(x)\,\mathrm{d}x=F(b)-F(a)$$

计算函数 $f(x)$ 在区间 $[a,b]$ 上的积分。这里，函数 $F(x)$ 是被积函数 $f(x)$ 的原函数，即 $F'(x)=f(x)$。但由于被积函数的原函数往往难以得出或比较复杂，采用牛顿-莱布尼兹公式计算积分可能不方便，因此有必要研究定积分的数值计算方法。

所谓数值积分就是用一个容易计算的积分来近似原积分。常常首先考虑用一个容易计算积分的函数来近似被积函数，使积分转化为一个简单积分，这也叫作近似积分。由积分中值定理可知，如果函数 $f(x)$ 在积分区间 $[a,b]$ 上连续，则在 $[a,b]$ 内存在一点 ξ，使 $\int_a^b f(x)\,\mathrm{d}x=f(\xi)(b-a)$ 成立。又根据定积分的定义 $\int_a^b f(x)\,\mathrm{d}x=\lim\limits_{\|\Delta x\|\to 0}\sum\limits_{i=0}^{n} f(\xi_i)\Delta x_i$ 可知，定积分是求和的极限值，显然可以把有限项的和作为定积分的近似值。

定义 (3.1)　数值求积方法是指，在积分区间 $[a,b]$ 上适当地选取一些点 $x_i\,(i=0,1,\cdots,n)$，用被积函数在这些点处值 $f(x_i)\,(i=0,1,\cdots,n)$ 的加权平均得到 $f(\xi)$，结合公式 $\int_a^b f(x)\,\mathrm{d}x=f(\xi)(b-a)$，构造出求积公式

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=0}^{n} A_i f(x_i) \tag{3.1}$$

通常称这种近似求积方法为机械求积法。其中，A_i 称为求积系数(或积分系数)，它是与被积函数 $f(x)$ 无关的常数；$x_i(i=0,1,\cdots,n)$ 称为积分节点；公式(3.1) 右边的和式

$$Q[f]=\sum_{i=0}^{n} A_i f(x_i) \tag{3.2}$$

称为求积公式；而

$$R[f]=\int_a^b f(x)\,\mathrm{d}x-\sum_{i=0}^{n} A_i f(x_i) \tag{3.3}$$

称为求积公式的余项。这样的求积公式也记为

$$I[f]=Q[f]+R[f] \tag{3.4}$$

数值积分就是用被积函数在积分区间内一些离散节点处值的线性组合来计算积分的近似值，把定积分的计算转化为函数值的计算，从而避免了应用牛顿-莱布尼兹公式需要求原函数的困难，并为借助计算机计算积分的近似值提供了简便可行的途径。

为了描述数值求积公式的近似程度，引入代数精度的概念。

定义(3.2)　如果数值求积公式(3.2) 对于次数不超过 m 的所有代数多项式都精确成立，而对某一个 $m+1$ 次代数多项式不精确成立，则称该数值求积公式具有m 次代数精度。

用定义(3.2) 直接判定数值求积公式的代数精度是不易操作的。依据多项式的性质和定积分的特点，可以证明定义(3.2) 等价于：要使数值求积公式(3.2) 具有 m 次代数精度，必须使它对于被积函数 $f(x)=1,x,x^2,\cdots,x^m$ 都精确成立，即要求

$$\sum_{i=0}^{n} A_i x_i^k=\frac{1}{k+1}(b^{k+1}-a^{k+1}) \quad (k=0,1,\cdots,m) \tag{3.5}$$

上式包括 $m+1$ 个公式，包含 $n+1$ 个节点 x_i 及 $n+1$ 个积分系数 A_i。如果事先选定了 x_i，并且取 $m=n$，则由上式可以求出 A_i，从而使求积公式至少具有 n 次代数精度。由此可见，构造数值求积公式实际上是求 x_i 与 A_i 的代数问题。

显然，求积公式的代数精度越高，就能对越多的被积函数准确成立，求积公式就具有更高的应用价值。

例 3.1　设求积公式

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \omega_0 f(-1)+\omega_1 f(0)+\omega_2 f(1)$$

试确定系数 ω_0,ω_1 和 ω_2，使求积公式的代数精度最高，并指出公式的实际代数精度。

解　令求积公式对被积函数 $f(x)=1,x,x^2$ 都精确成立，得到系数 ω_0,ω_1 和 ω_2 满足的线性方程组

$$\begin{cases}\omega_0+\omega_1+\omega_2=\int_{-1}^{1} 1\mathrm{d}x=2\\ -\omega_0+\omega_2=\int_{-1}^{1} x\mathrm{d}x=0\\ \omega_0+\omega_2=\int_{-1}^{1} x^2\mathrm{d}x=\dfrac{2}{3}\end{cases}$$

解得

$$\omega_0=\frac{1}{3},\quad \omega_1=\frac{4}{3},\quad \omega_2=\frac{1}{3}$$

因此求积公式是

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \frac{1}{3}f(-1)+\frac{4}{3}f(0)+\frac{1}{3}f(1)$$

对于函数 $f(x)=x^3$,上述积分公式给出

$$\frac{1}{3}(-1)^3+\frac{4}{3}(0)^3+\frac{1}{3}(1)^3=0=\int_{-1}^{1} x^3\,\mathrm{d}x$$

显然积分成立。但当取函数 $f(x)=x^4$ 时,有

$$\frac{1}{3}(-1)^4+\frac{4}{3}(0)^4+\frac{1}{3}(1)^4=\frac{2}{3}\neq\frac{2}{5}=\int_{-1}^{1} x^4\,\mathrm{d}x$$

积分公式不成立。因此,积分公式实际具有三次代数精度。

3.2 插值型求积公式

定义(3.3) 在积分区间 $[a,b]$ 上选取插值节点 $x_i(i=0,1,\cdots,n)$,满足

$$a=x_0<x_1<\cdots<x_n=b$$

当被积函数 $f(x)$ 在插值节点 $x_i(i=0,1,\cdots,n)$ 处的值 $f(x_i)$ $(i=0,1,\cdots,n)$ 已知时,就可以构造插值函数 $\varphi(x)$,用 $\varphi(x)$ 代替被积函数 $f(x)$ 来近似计算积分,亦即以 $\int_a^b\varphi(x)\,\mathrm{d}x$ 作为 $\int_a^b f(x)\,\mathrm{d}x$ 的近似值。这样构造出的求积公式

$$Q_n[f]=\int_a^b\varphi(x)\,\mathrm{d}x=\sum_{i=0}^{n}A_if(x_i) \tag{3.6}$$

称为插值型求积公式。

若采用拉格朗日插值公式,有

$$f(x)=L_n(x)+R_n(x) \tag{3.7}$$

式中,拉格朗日插值多项式 $L_n(x)$ 及其余项 $R_n(x)$ 分别是

$$L_n(x)=\sum_{i=0}^{n}l_i(x)f(x_i)$$

$$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x-x_i)\quad(\xi\in(a,b))$$

其中,插值基函数

$$l_i(x)=\prod_{\substack{k=0\\k\neq i}}^{n}\frac{(x-x_k)}{(x_i-x_k)}\quad(i=0,1,\cdots,n)$$

对插值基函数 $l_i(x)$ 积分就得到求积系数

$$A_i=\int_a^b l_i(x)\,\mathrm{d}x=\int_a^b\prod_{\substack{k=0\\k\neq i}}^{n}\frac{(x-x_k)}{(x_i-x_k)}\mathrm{d}x\quad(i=0,1,\cdots,n) \tag{3.8}$$

由插值余项公式可得插值型求积公式的余项是

$$R_n[f]=\int_a^b R_n(x)\,dx=\int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x-x_i)\,dx \tag{3.9}$$

对于插值型求积公式 $Q_n[f]=\sum_{i=0}^{n}A_if(x_i)$，当被积函数 $f(x)$ 是次数不高于 n 的代数多项式时，由于 $f^{(n+1)}(x)\equiv 0$，所以求积公式的余项 $R_n[f]\equiv 0$，即多项式插值型求积公式(3.6)对任何次数不超过 n 次的代数多项式被积函数均精确成立，所以含有 $n+1$ 个节点的多项式插值型求积公式至少具有 n 次代数精度。

定理(3.1)　形如 $I_n=\sum_{i=0}^{n}A_if(x_i)$ 的求积公式至少有 n 次代数精度的充分必要条件是它是多项式插值型的。

当 $f(x)=1$ 时，插值型求积公式(3.6)也精确成立，因而有

$$\sum_{i=0}^{n}A_i=b-a$$

即插值型求积公式所有积分系数的和等于积分区间的长度。

3.3　牛顿-柯特斯积分公式

牛顿-柯特斯(Newton-Cotes)求积公式就是等距节点情况下的多项式插值型求积公式。

定义(3.4)　对积分区间 $[a,b]$ 进行 n 等分，步长 $h=(b-a)/n$，积分节点是等距节点 $x_i=a+ih\ (i=0,1,\cdots,n)$，且积分节点处的被积函数值 $f(x_i)\ (i=0,1,\cdots,n)$ 已知，则以这些积分节点为插值节点建立的插值型求积公式就是 n 阶牛顿-柯特斯积分公式，记为 N－C 公式。它是

$$I_n[f]=(b-a)\sum_{i=0}^{n}C_i^{(n)}f(x_i) \tag{3.10}$$

其中，柯特斯系数是

$$C_i^{(n)}=\frac{A_i}{b-a}=\frac{1}{b-a}\int_a^b\prod_{\substack{j=0\\j\neq i}}^{n}\left(\frac{x-x_j}{x_i-x_j}\right)dx$$

为了简化公式，作变换 $x=a+sh$，则有

$$dx=h\,ds,\quad x_i-x_j=(i-j)h,\quad x-x_j=(s-j)h\quad (j=0,1,\cdots,n)$$

从而

$$C_i^{(n)}=\frac{h}{b-a}\int_0^n\prod_{\substack{j=0\\j\neq i}}^{n}\left(\frac{s-j}{i-j}\right)ds=\frac{1}{n}\int_0^n\prod_{\substack{j=0\\j\neq i}}^{n}\left(\frac{s-j}{i-j}\right)ds=$$
$$\frac{(-1)^{n-i}}{n\cdot i!(n-i)!}\int_0^n\prod_{\substack{j=0\\j\neq i}}^{n}(s-j)\,ds\quad (i=0,1,\cdots,n) \tag{3.11}$$

显然，柯特斯系数 $C_i^{(n)}$ 只与 n 和 i 有关，而与 a,b 及被积函数 $f(x)$ 无关，因此可以单独计算出并制成表格(见表 3.1)，以便需要时查用。

下面给出几种常用的低阶牛顿-柯特斯求积公式。

1. 梯形公式

当 $n=1$ 时，$x_0=a$，$x_1=b$，由式(3.11)或表 3.1 可得柯特斯系数为

$$C_0^1 = \frac{1}{2}, \quad C_1^1 = \frac{1}{2}$$

于是

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{2}[f(a)+f(b)]$$

称为梯形求积公式，简称梯形公式，记作

$$T[f] = \frac{b-a}{2}[f(a)+f(b)] \tag{3.12}$$

表 3.1

$C_i^{(n)}$		$i(i=0,1,\cdots,n)$							
		0	1	2	3	4	5	6	7
n	1	$\frac{1}{2}$	$\frac{1}{2}$						
	2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$					
	3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$				
	4	$\frac{7}{90}$	$\frac{16}{45}$	$\frac{2}{15}$	$\frac{16}{45}$	$\frac{7}{90}$			
	5	$\frac{19}{288}$	$\frac{25}{96}$	$\frac{25}{144}$	$\frac{25}{144}$	$\frac{25}{96}$	$\frac{19}{288}$		
	6	$\frac{41}{840}$	$\frac{9}{35}$	$\frac{9}{280}$	$\frac{34}{105}$	$\frac{9}{280}$	$\frac{9}{35}$	$\frac{41}{840}$	
	7	$\frac{751}{17\,280}$	$\frac{3\,577}{17\,280}$	$\frac{1\,323}{17\,280}$	$\frac{2\,989}{17\,280}$	$\frac{2\,989}{17\,280}$	$\frac{1\,323}{17\,280}$	$\frac{3\,577}{17\,280}$	$\frac{751}{17\,280}$

如图 3.1 所示，梯形求积公式(3.12)的几何意义是，用梯形面积来近似被积函数曲线下边的面积。

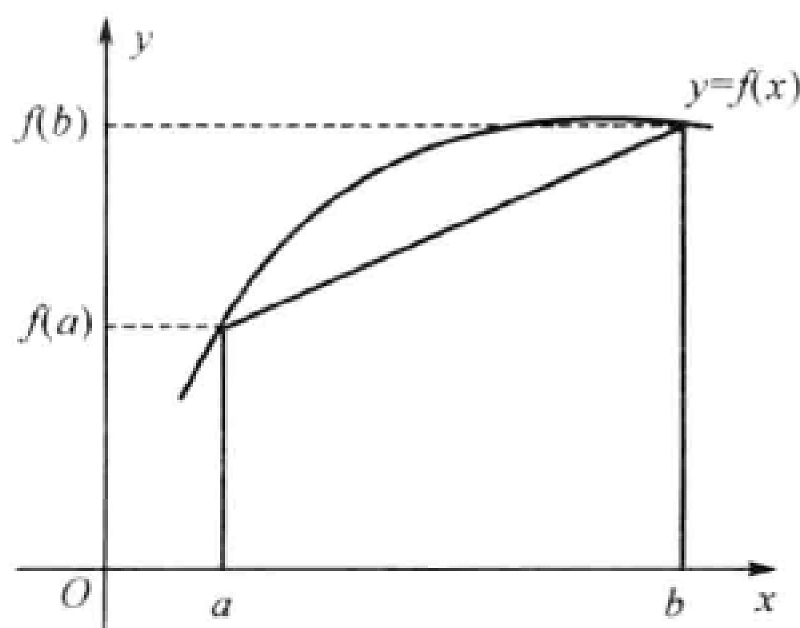


图 3.1

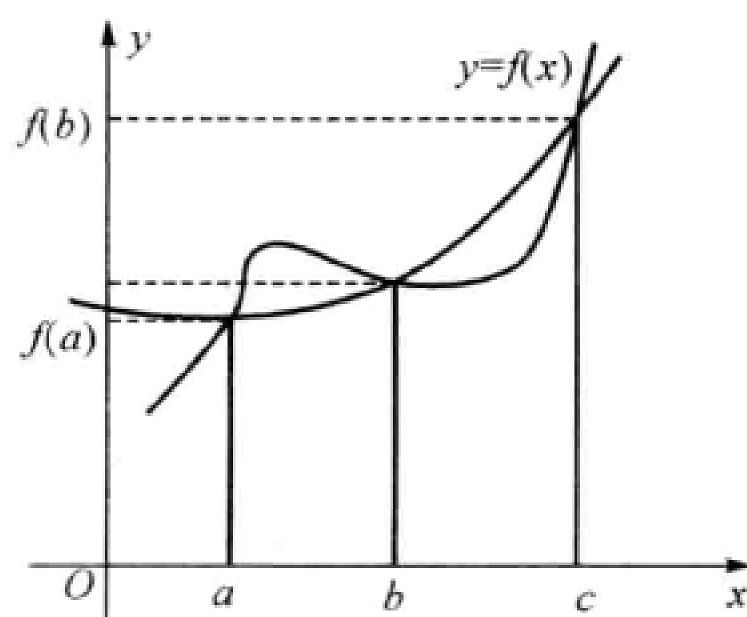


图 3.2

2. 辛普森公式

当 $n=2$ 时，插值节点是 $x_0=a, x_1=(a+b)/2, x_2=b$，由公式(3.11)或表 3.1 可得柯特斯系数为

$$C_0^2=\frac{1}{6},\quad C_1^2=\frac{4}{6},\quad C_2^2=\frac{1}{6}$$

于是

$$\int_a^b f(x)\,\mathrm{d}x\approx\frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right]$$

称为辛普森(Simpson)公式(或抛物线公式),记作

$$S[f]=\frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right]\tag{3.13}$$

其几何意义是,用抛物线下边的面积去近似曲线下边的面积,如图 3.2 所示。

3. 柯特斯公式

当 $n=4$ 时,牛顿-柯特斯积分公式(3.10)表示为

$$\int_a^b f(x)\,\mathrm{d}x\approx\frac{b-a}{90}[7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)]$$

其中,$x_i=a+i(b-a)/4(i=0,1,2,3,4)$。上式称为柯特斯公式,记作

$$C[f]=\frac{b-a}{90}[7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)]\tag{3.14}$$

例 3.2　试用梯形公式、辛普森公式和柯特斯公式分别计算积分 $\int_{0.5}^{1}\sqrt{x}\,\mathrm{d}x$(结果取 5 位有效数字)。

解　用梯形公式(3.12)计算可得

$$\int_{0.5}^{1}\sqrt{x}\,\mathrm{d}x\approx\frac{0.5}{2}(\sqrt{0.5}+\sqrt{1})=0.426\ 78$$

用辛普森公式(3.13)计算可得

$$\int_{0.5}^{1}\sqrt{x}\,\mathrm{d}x\approx\frac{0.5}{6}(\sqrt{0.5}+4\sqrt{0.75}+\sqrt{1})=0.430\ 93$$

用柯特斯公式(3.14)计算可得

$$\int_{0.5}^{1}\sqrt{x}\,\mathrm{d}x\approx\frac{0.5}{90}(7\sqrt{0.5}+32\sqrt{0.625}+12\sqrt{0.75}+32\sqrt{0.875}+7\sqrt{1})=0.430\ 96$$

积分的准确值是

$$\int_{0.5}^{1}\sqrt{x}\,\mathrm{d}x=\frac{2}{3}x^{\frac{3}{2}}\Big|_{0.5}^{1}=0.430\ 96$$

显然,从梯形公式、辛普森公式到柯特斯公式,计算精度依次提高。

牛顿-柯特斯求积公式的余项仍由式(3.9)表示。常用的低阶牛顿-柯特斯求积公式——梯形公式、辛普森公式和柯特斯公式的余项分别由下列几个定理给出。

定理(3.2)　设函数 $f(x)$ 在区间 $[a,b]$ 上具有二阶连续导数,则梯形公式(3.12)的余项为

$$R_T[f]=-\frac{(b-a)^3}{12}f''(\eta)\quad(a<\eta<b)\tag{3.15}$$

证明　在余项公式(3.9)中,令 $n=1$,得到梯形公式的余项为

$$R_T[f]=\int_a^b\frac{f''(\xi)}{2}(x-a)(x-b)\,\mathrm{d}x\quad(a<\xi<b)$$

由于 $(x-a)(x-b)$ 在区间 $[a,b]$ 上恒为负,$f''(x)$ 在 $[a,b]$ 上连续,由积分中值定理,存在

$\eta \in (a,b)$，使

$$R_T[f]=\frac{f''(\eta)}{2}\int_a^b (x-a)(x-b)\,\mathrm{d}x=-\frac{(b-a)^3}{12}f''(\eta)$$

定理(3.3)　设函数 $f(x)$ 在区间 $[a,b]$ 上具有连续的四阶导数，则辛普森公式(3.13)的余项为

$$R_S[f]=-\frac{(b-a)^5}{2\ 880}f^{(4)}(\eta)=-\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\eta)\quad (a<\eta<b) \tag{3.16}$$

定理(3.4)　设函数 $f(x)$ 在区间 $[a,b]$ 上具有连续的六阶导数，则柯特斯公式(3.14)的余项为

$$R_C[f]=-\frac{8}{945}\left(\frac{b-a}{4}\right)^7 f^{(6)}(\eta)\quad (a<\eta<b) \tag{3.17}$$

从以上三个定理可以看出，梯形公式只有一次代数精度，辛普森公式有三次代数精度，而柯特斯公式具有五次代数精度。关于一般情况下牛顿-柯特斯求积公式的代数精度有定理：

定理(3.5)　对于 n 阶牛顿-柯特斯求积公式 $I_n[f]=(b-a)\sum\limits_{i=0}^{n}C_i^{(n)}f(x_i)$，当 n 为奇数时，至少具有 n 次代数精度；当 n 为偶数时，至少具有 $n+1$ 次代数精度。

可以证明：当 $n\leqslant 7$ 时，牛顿-柯特斯积分公式(3.10)对被积函数在积分节点处值的误差不会放大，数值积分计算过程是稳定的；而当 $n\geqslant 8$ 时，牛顿-柯特斯积分公式数值计算过程的稳定性不能保证，故 $n\geqslant 8$ 的牛顿-柯特斯求积公式通常不用。

程序(3.1)　牛顿-柯特斯积分的 MATLAB 程序。

程序任务：用牛顿-柯特斯积分公式计算给定函数在区间 $[a,b]$ 上的积分。

```
function i = newtoncotes(funame,a,b,n)
% 输入:funame—— 被积函数
%        a—— 积分区间下限
%        b—— 积分区间上限
%        n—— 积分区间的等分数:n = 1 时采用梯形公式计算积分,n = 2 时采用辛普森公式计算积分,
%            n = 4 时采用柯特斯公式计算积分
% 输出:i—— 数值积分计算结果
fa = feval(funame, a); fb = feval(funame, b);                    %f(a) 和 f(b)
if n == 1
    i = 0.5 * (b - a) * (fa + fb);                               % 梯形公式(3.12)
elseif n == 2
    i = ((b - a)/6) * (fa + 4 * feval(funame, (a + b)/2) + fb);   % 辛普森公式(3.13)
elseif n == 4
    i = ((b - a)/90) * (7 * (fa + fb) + 32 * ( feval(funame, a + (b - a)/4) + feval(funame, a...
    + 3 * (b - a)/4)) + 12 * feval(funame, (b - a)/2));           % 柯特斯公式(3.14)
else
    error(' 区间划分数必须等于 1,2 或 4');
end
```

在 MATLAB 命令窗口输入：

```
>> i = newtoncotes(inline('exp(x)'),0,0.5,1)  % 采用梯形公式计算积分∫_0^0.5 exp(x)dx
```

输出结果：

```
i =0.66218031767503
>> i = newtoncotes(inline('exp(x)'),0,0.5,2)  % 采用辛普森公式计算积分
```

输出结果：

```
i =0.64873524478759
>> i = newtoncotes(inline('exp(x)'),0,0.5,4)  % 采用柯特斯公式计算积分
```

输出结果：

```
i =0.64872127589486
```

积分$\int_0^{0.5}\exp(x)\mathrm{d}x = 0.648\ 721\ 270\ 700\ 13$。显然，从梯形公式到柯特斯公式，计算值越来越准确。

3.4　复化求积方法

为了同时克服高阶插值型积分可能出现的数值计算不稳定和低阶插值型数值积分计算结果精度低这两方面的不足，人们提出了复化求积方法。复化求积方法就是先把积分区间细分，并在细分得到的每个小区间上使用低阶插值型求积公式计算积分，然后把所有小区间上的积分值加起来得出原积分的近似值。

3.4.1　复化求积公式

1. 复化梯形公式

把积分区间$[a,b]$进行n等分，步长$h=(b-a)/n$，积分节点$x_i=a+ih\ (i=0,1,\cdots,n)$，在子区间$[x_i,x_{i+1}]$上使用梯形公式

$$\int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x \approx \frac{h}{2}[f(x_i)+f(x_{i+1})] \quad (i=0,1,\cdots,n-1)$$

对所有子区间求和可得

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x \approx \frac{h}{2}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right]$$

记作

$$T_n = \frac{h}{2}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right] \tag{3.18}$$

式(3.18)称为复化梯形公式。

定理(3.6)　若$f(x)$在区间$[a,b]$上具有连续的二阶导数，则复化梯形公式(3.18)的余项为

$$R_{T_n}[f] = -\frac{(b-a)^3}{12n^2}f''(\eta) \quad (a<\eta<b) \tag{3.19}$$

证明　由梯形公式的余项式(3.15)，在子区间$[x_i,x_{i+1}]$上用梯形公式计算的$\int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x$余项为$-h^3 f''(\eta_i)/12\ (x_i<\eta_i<x_{i+1})$，将$n$个子区间的余项求和就得到复化梯

形公式在区间$[a,b]$上的余项,即

$$R_{T_n}[f]=-\frac{h^3}{12}[f''(\eta_0)+f''(\eta_1)+\cdots+f''(\eta_{n-1})]$$

由于$f''(x)$在$[a,b]$上连续,因此在$[a,b]$内存在一点η,使

$$f''(\eta)=\frac{1}{n}[f''(\eta_0)+f''(\eta_1)+\cdots+f''(\eta_{n-1})]$$

于是有

$$R_{T_n}[f]=-\frac{(b-a)^3}{12n^2}f''(\eta)$$

若$M=\max\limits_{x\in(a,b)}|f''(x)|$,则有

$$|R_{T_n}[f]|\leqslant\frac{(b-a)^3}{12n^2}M \tag{3.20}$$

程序(3.2) 复化梯形公式的 MATLAB 程序。

程序任务:用复化梯形公式计算给定函数在区间上的积分。

```
function i = trapi(x,y)
% 输入:x—— 自变量的等距节点向量
%       y—— 被积函数在节点处的值向量
% 输出:i—— 数值积分计算结果
n = length(x); m = length(y);
if n ~= m
   error('向量 x 与向量 y 的长度必须相同'); end
h = (x(n) - x(1))/(n - 1);
a = [1,2. * ones(1,n - 2),1];        % 建立系数向量
i = h/2 * sum(a. * y);               % 计算积分值,公式(3.18)
```

在 MATLAB 命令窗口输入:

```
>> x=0:0.01:1;                       % 建立积分变量等距节点坐标向量
>> y=1 + exp(- x). * sin(4. * x);    % 计算被积函数在节点处的值向量
>> i = trapi(x,y)                    % 调用程序(3.2)计算积分∫_0^1[1+e^{-x}sin(4x)]dx
```

输出结果:

```
i =1.30821157529417
```

积分的准确值$\int_0^1[1+\mathrm{e}^{-x}\sin(4x)]\mathrm{d}x=\frac{21e-4\cos4-\sin4}{17\mathrm{e}}=1.308\ 250\ 604\ 642\ 6$。

2. 复化辛普森公式

把积分区间$[a,b]$进行$2m$等分,分点坐标是$x_j=a+jh\ (j=0,1,\cdots,2m)$,相邻分点之间的距离为$h=(b-a)/2m$。在子区间$[x_{2i},x_{2i+2}]\ (i=0,1,\cdots,m-1)$上使用辛普森公式,有

$$\int_{x_{2i}}^{x_{2i+2}}f(x)\mathrm{d}x\approx\frac{h}{3}[f(x_{2i})+4f(x_{2i+1})+f(x_{2i+2})]\quad(i=0,1,\cdots,m-1)$$

对所有子区间求和可得

$$\int_a^b f(x)\mathrm{d}x=\sum_{i=0}^{m-1}\int_{x_{2i}}^{x_{2i+2}}f(x)\mathrm{d}x\approx\frac{h}{3}\left[f(a)+4\sum_{i=0}^{m-1}f(x_{2i+1})+2\sum_{i=1}^{m-1}f(x_{2i})+f(b)\right]$$

记作

$$S_m=\frac{h}{3}\Big[f(a)+4\sum_{i=0}^{m-1}f(x_{2i+1})+2\sum_{i=1}^{m-1}f(x_{2i})+f(b)\Big] \tag{3.21}$$

称为复化辛普森公式。

定理(3.7)　若 $f(x)$ 在区间 $[a,b]$ 上具有连续的四阶导数，则复化辛普森公式(3.21) 的余项为

$$R_{S_{2m}}[f]=-\frac{b-a}{180}h^4f^{(4)}(\eta)\quad(a<\eta<b) \tag{3.22}$$

程序(3.3)　复化辛普森公式的 MATLAB 程序。

程序任务：用复化辛普森公式计算给定函数的积分。

```
function i = simpi(x,y)
% 输入：x—— 自变量的等距节点向量
%      y—— 被积函数在节点处的值向量
% 输出：i—— 数值积分计算结果
n = length(x); m = length(y);
if n ~= m error('向量 x 与向量 y 的长度必须相同'); end
if rem(n-1,2) ~= 0
    i = trapi(x,y);              % 当区间划分数不是偶数时，转到程序(3.2)，采用复化梯形积分
    return;
end
N = (n-1)/2; h = (x(n) - x(1))/N;
a = zeros(1,n);
for k = 1:N                      % 建立系数向量
    a(2*k-1) = a(2*k-1)+1;
    a(2*k) = a(2*k)+4;
    a(2*k+1) = a(2*k+1)+1;
end
i = h/6 * sum(a.*y);             % 计算积分值，公式(3.21)
```

在 MATLAB 命令窗口输入：

```
>> x=0:0.01:1;                   % 建立自变量等距节点向量
>> y=1+exp(-x).*sin(4.*x);       % 计算被积函数在节点处的值向量
>> i=simpi(x,y)                  % 调用程序(3.3) 计算积分∫_0^1 [1+e^(-x) sin(4x)]dx
```

输出结果：

```
i =1.308 250 607 499 39
```

3. 复化柯特斯公式

把积分区间 $[a,b]$ 进行 $4m$ 等分，步长 $h=(b-a)/4m$，积分节点 $x_i=a+jh\ (j=0,1,\cdots,4m)$，在子区间 $[x_{4i},x_{4i+4}]\ (i=0,1,\cdots,m-1)$ 上使用柯特斯公式，有

$$\int_{x_{4i}}^{x_{4i+4}}f(x)\,\mathrm{d}x\approx\frac{b-a}{90m}[7f(x_{4i})+32f(x_{4i+1})+12f(x_{4i+2})+32f(x_{4i+3})+7f(x_{4i+4})]$$

对所有区间求和可得复化科特斯公式

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{90m}\Big[7f(a)+32\sum_{i=0}^{m-1}f(x_{4i+1})+12\sum_{i=0}^{m-1}f(x_{4i+2})+ 32\sum_{i=0}^{m-1}f(x_{4i+3})+14\sum_{i=0}^{m-2}f(x_{4i+4})+7f(b)\Big]$$

记作

$$C_m=\frac{b-a}{90m}\Big[7f(a)+32\sum_{i=0}^{m-1}f(x_{4i+1})+12\sum_{i=0}^{m-1}f(x_{4i+2})+ 32\sum_{i=0}^{m-1}f(x_{4i+3})+14\sum_{i=0}^{m-2}f(x_{4i+4})+7f(b)\Big] \tag{3.23}$$

定理(3.8) 若 $f(x)$ 在区间 $[a,b]$ 上具有连续的六阶导数，则复化柯特斯公式(3.23) 的余项为

$$R_{C_{4m}}[f]=-\frac{2(b-a)}{945}h^6 f^{(6)}(\eta) \quad (a<\eta<b) \tag{3.24}$$

程序(3.4) 复化柯特斯公式的 MATLAB 程序。

程序任务：用复化柯特斯公式计算给定函数在区间上的积分。

```
function i = cotes(x,y)
% 输入:x—— 自变量的等距节点向量
%      y—— 被积函数在节点处的值向量
% 输出:i—— 数值积分计算结果
n = length(x); m = length(y);
if n ~= m
    error('向量 x 与向量 y 的长度必须相同');
end
if rem(n-1,4) ~= 0
    i = simpi(x,y);      % 当区间划分数不是 4 的倍数时,转到程序(3.3),采用复化辛普森方法积分
    return;
end
N = (n-1)/4; h = (x(n) - x(1))/N;
a = zeros(1,n);
for k = 1:N                  % 建立系数向量
    a(4*k-3) = a(4*k-3) + 7;
    a(4*k-2) = a(4*k-2) + 32;
    a(4*k-1) = a(4*k-1) + 12;
    a(4*k) = a(4*k) + 32;
    a(4*k+1) = a(4*k+1) + 7;
end
i = h/90 * sum(a.*y);  % 计算积分值,公式(3.23)
```

在 MATLAB 命令窗口输入：

```
>> x=0:0.01:1;                    % 建立自变量等距节点向量
```

```
≫ y = 1 + exp(- x). * sin(4. * x); % 计算被积函数在节点处的值向量
≫ i = cotes(x,y)                    % 调用程序(3.4) 计算积分∫_0^1 [1 + e^(-x) sin(4x)] dx
```

输出结果：

```
i = 1.308 250 604 642 27
```

例 3.3　分别用复化梯形积分、复化辛普森积分和复化柯特斯积分根据表 3.2 计算积分 $I=\int_0^1 \frac{\sin x}{x}\mathrm{d}x$ 的近似值。

表　3.2

x	0	1/8	1/4	3/8	1/2	5/8	3/4	7/8	1
$f(x)=\frac{\sin x}{x}$	1	0.997 397 8	0.989 615 8	0.976 726 7	0.958 851 0	0.936 155 6	0.908 851 6	0.877 192 5	0.841 470 9

解　(1) 用复化梯形积分计算。取 $n=8$，将区间 $[0,1]$ 分成 8 等份，$h=1/8$，由复化梯形公式(3.18) 得

$$T_8=\frac{1}{2}\times\frac{1}{8}\left[f(0)+2\sum_{i=1}^{7}f\left(\frac{i}{8}\right)+f(1)\right]\approx 0.945\ 690\ 9$$

(2) 用复化辛普森公式计算。取 $m=4$，将区间 $[0,1]$ 分成 $2m=8$ 等份，$h=1/8$，由复化辛普森公式(3.21) 得

$$S_4=\frac{1}{3}\times\frac{1}{8}\left[f(0)+4\sum_{i=0}^{3}f\left(\frac{2i+1}{8}\right)+2\sum_{i=1}^{3}f\left(\frac{i}{4}\right)+f(1)\right]\approx 0.946\ 083\ 2$$

(3) 用复化柯特斯公式计算。取 $m=2$，将区间 $[0,1]$ 分成 $4m=8$ 等份，$h=1/8$，由复化柯特斯公式(3.23) 得

$$C_2=\frac{1}{90\times 2}\left[7f(0)+32\sum_{i=0}^{1}f\left(\frac{4i+1}{8}\right)+12\sum_{i=0}^{1}f\left(\frac{2i+1}{4}\right)+\right.$$
$$\left.32\sum_{i=0}^{1}f\left(\frac{4i+3}{8}\right)+14f\left(\frac{1}{2}\right)+7f(1)\right]\approx 0.945\ 694\ 1$$

可以证明，当步长 $h\rightarrow 0$ 时，复化梯形公式、复化辛普森公式和复化柯特斯公式均收敛到 $\int_a^b f(x)\,\mathrm{d}x$，而且收敛速度一个比一个快，计算都是数值稳定的。

3.4.2　截断误差的事后估计与步长的自动选择

在使用复化积分公式计算积分近似值以前，必须确定所采用的积分步长。若被积函数的高阶导数容易估计，则能根据复化积分的余项公式，较为方便地选择满足结果误差要求的积分步长。但当被积函数的高阶导数不知道或很难估计时，就无法用余项公式确定积分步长。这里介绍一种在积分计算过程中能适时确定积分步长的变步长求积方法。

变步长求积方法也称为区间逐次分半法。其基本思路是：在对积分区间逐次加倍等分并采用复化求积公式计算积分的过程中，用连续两次积分计算结果之差的绝对值 $|I_{2n}-I_n|$ 小于给定的允许误差 ε 作为终止计算的依据，并取 I_{2n} 为积分的最终近似值。由于在整个计算过程

中积分步长是逐步变化的，因此称为变步长求积方法。

1. 复化梯形公式

把积分区间$[a,b]$进行n等分，用复化梯形方法计算的积分近似值为T_n，积分的准确值记作I，则从式(3.19)可得

$$I-T_n=-\frac{(b-a)}{12}\left(\frac{b-a}{n}\right)^2 f''(\eta_1) \quad (a<\eta_1<b)$$

再将区间$[a,b]$进行$2n$等分，用复化梯形方法计算的积分近似值为T_{2n}，有

$$I-T_{2n}=-\frac{(b-a)}{12}\left(\frac{b-a}{2n}\right)^2 f''(\eta_2) \quad (a<\eta_2<b)$$

若$f''(x)$在区间$[a,b]$上连续并且变化不大，即$f''(\eta_1)\approx f''(\eta_2)$，从前两式可以得出

$$\frac{I-T_{2n}}{I-T_n}\approx\left(\frac{1}{2}\right)^2$$

说明，当步长减小一半时，截断误差减小到1/4。将上式变形后就是

$$I\approx T_{2n}+\frac{1}{3}(T_{2n}-T_n) \tag{3.25}$$

或

$$I-T_{2n}\approx\frac{1}{3}(T_{2n}-T_n) \tag{3.26}$$

表明，若以T_{2n}作为I的近似值，则截断误差约为$\frac{1}{3}(T_{2n}-T_n)$，因此在对区间逐次加倍等分并计算积分的过程中，可以用$|T_{2n}-T_n|$来估计截断误差并据此确定积分步长。具体方法是：先计算T_n和T_{2n}，如果$|T_{2n}-T_n|<\varepsilon$，则停止计算，取T_{2n}为积分的近似值；否则，将每个小区间再分半后计算T_{4n}，并检查$|T_{4n}-T_{2n}|<\varepsilon$是否成立；如此反复计算，直到得出满足误差要求的结果为止。

把这种用连续两次积分计算结果之差来估计截断误差的方法称为截断误差的事后估计法。

2. 复化辛普森公式和复化柯特斯公式

若$f^{(4)}(x)$在$[a,b]$上连续并且变化不大，同样可以推导出

$$\frac{I-S_{2n}}{I-S_n}\approx\left(\frac{1}{2}\right)^4$$

整理后就是

$$I\approx S_{2n}+\frac{1}{15}(S_{2n}-S_n) \tag{3.27}$$

或

$$I-S_{2n}\approx\frac{1}{15}(S_{2n}-S_n) \tag{3.28}$$

如果$f^{(6)}(x)$在$[a,b]$上连续并且变化不大，也能够证明

$$\frac{I-C_{2n}}{I-C_n}\approx\left(\frac{1}{2}\right)^6$$

整理后就是

$$I\approx C_{2n}+\frac{1}{63}(C_{2n}-C_n) \tag{3.29}$$

或

$$I - C_{2n} \approx \frac{1}{63}(C_{2n} - C_n) \tag{3.30}$$

因此,对复化辛普森公式和复化柯特斯公式,也可以如处理复化梯形公式那样,采用变步长积分法并对截断误差进行事后估计。

3.4.3 复化梯形积分的递推公式

用复化梯形公式计算 T_n 时,先对积分区间 $[a,b]$ 进行 n 等分,需要计算被积函数在 $n+1$ 个积分节点处的值 $f(a+i(b-a)/n)$ $(i=0,1,\cdots,n)$。之后,若再用复化梯形公式计算 T_{2n},则要对积分区间 $[a,b]$ 进行 $2n$ 等分,并计算被积函数在 $2n+1$ 个积分节点处的值 $f(a+i(b-a)/2n)$ $(i=0,1,\cdots,2n)$。显然,计算 T_{2n} 时,对在计算 T_n 过程中已经计算过的 $n+1$ 个节点处的函数值 $f(a+i(b-a)/2n)$ $(i=0,2,4,\cdots,2n)$ 再次进行了计算,这显然增加了运算量,降低了计算速度。为了避免这种重复计算,有必要分析 T_{2n} 与 T_n 的关系。

令 $h_{2n}=(b-a)/2n$,则由公式(3.18)有

$$T_{2n} = \frac{b-a}{4n}\left[f(a) + 2\sum_{i=1}^{2n-1} f\left(a + i\frac{b-a}{2n}\right) + f(b)\right] =$$

$$\frac{h_{2n}}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(a+2ih_{2n}) + 2\sum_{i=1}^{n} f(a+(2i-1)h_{2n}) + f(b)\right] =$$

$$\frac{h_{2n}}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(a+2ih_{2n}) + f(b)\right] + h_{2n}\sum_{i=1}^{n} f(a+(2i-1)h_{2n}) =$$

$$\frac{1}{2}\left\{\frac{b-a}{2n}\left[f(a) + 2\sum_{i=1}^{n-1} f\left(a + i\frac{b-a}{n}\right) + f(b)\right]\right\} + h_{2n}\sum_{i=1}^{n} f(a+(2i-1)h_{2n})$$

得到复化梯形积分的递推公式

$$T_{2n} = \frac{1}{2}T_n + h_{2n}\sum_{i=1}^{n} f(a+(2i-1)h_{2n}) \tag{3.31}$$

可见,求出 T_n 后再计算 T_{2n} 时,只要计算 n 个新增积分节点上的被积函数值 $f(a+i(b-a)/2n)$ $(i=1,3,\cdots,2n-1)$ 即可。

为了便于编程,将复化梯形公式变步长积分过程的递推公式写成

$$T_1 = \frac{b-a}{2}[f(a)+f(b)]$$

$$T_{2^k} = \frac{1}{2}T_{2^{k-1}} + \frac{b-a}{2^k}\sum_{i=1}^{2^{k-1}} f\left(a+(2i-1)\frac{b-a}{2^k}\right) \quad (k=1,2,3,\cdots) \tag{3.32}$$

这里 T_{2^k} 表示将区间 $[a,b]$ 进行 2^k 等分时的复化梯形积分值。

对复化辛普森公式和复化柯特斯公式也能建立类似的递推公式。

例 3.4 用复化梯形积分的递推公式计算 $\pi=\int_0^1 \frac{4}{1+x^2}dx$ 的近似值,要求结果误差的绝对值 $\varepsilon \leqslant 10^{-5}$。

解 按照式(3.32),先计算 T_1:

$$T_1 = \frac{1}{2}[f(0)+f(1)] = 3.000\ 000$$

然后计算 T_2：

$$T_2=\frac{1}{2}T_1+\frac{1}{2}f\left(\frac{1}{2}\right)=3.100\ 000$$

再计算 T_4：

$$T_4=\frac{1}{2}T_2+\frac{1}{4}\left[f\left(\frac{1}{4}\right)+f\left(\frac{3}{4}\right)\right]=3.131\ 176$$

……

计算结果列入表 3.3 中。

表 3.3

k	T_{2^k}	k	T_{2^k}
0	3.000 000	5	3.141 430
1	3.100 000	6	3.141 552
2	3.131 176	7	3.141 582
3	3.138 988	8	3.141 590
4	3.140 942	9	3.141 592

根据表中数据有 $|T_{2^8}-T_{2^7}|=0.000\ 008<10^{-5}$，所以积分近似值取 $T_{2^8}=3.141\ 590$。

程序(3.5)　变步长梯形求积公式的 MATLAB 程序。

程序任务：用变步长梯形求积方法计算给定函数在区间上的积分。

```
function I = changtrap (funame,a,b,ep)
% 输入:funame—— 被积函数
%       a—— 积分区间下限
%       b—— 积分区间上限
%       ep—— 计算精度(默认值为 1e-5)
% 输出:I—— 数值积分计算结果
if nargin < 4 ep = 1e-5; end
N = 1; h = b-a;
T = h/2 * (feval(funame,a) + feval(funame,b));      % 计算 T1
while 1
    h = h/2; I = T/2;
    for k = 1:N
        I = I+ h. * feval(funame,a + (2 * k - 1) * h);% 公式(3.32)
    end
    if abs(I-T) < ep
        break;
    end
    N = 2 * N;
    T = I;
end
```

在 MATLAB 命令窗口输入：

```
>> I=changtrap (inline('1+exp(-x).*sin(4.*x)','x'),0,1,1e-8)
```

% 调用程序(3.5) 计算积分 $\int_0^1 [1+e^{-x}\sin(4x)]dx$

输出结果：

I=1.308 250 603 188 74

3.5 龙贝格方法

为了提高数值积分的收敛速度，把复化梯形公式计算的结果 T_n 和 T_{2n} 按一定方式线性组合，生成比复化梯形公式计算精度高的复化辛普森公式的计算结果 S_n；再把 S_n 和 S_{2n} 按一定方式线形组合，生成比复化辛普森公式计算精度高的复化柯特斯公式的计算结果 C_n；最后把 C_n 和 C_{2n} 按一定方式线形组合，生成比复化柯特斯公式计算精度高的龙贝格(Romberg) 公式计算结果 R_n。这种加速方法叫作龙贝格方法。

公式(3.25) 指出

$$I \approx T_{2n} + \frac{1}{3}(T_{2n} - T_n)$$

记

$$\overline{T} = T_{2n} + \frac{1}{3}(T_{2n} - T_n) = \frac{4}{3}T_{2n} - \frac{1}{3}T_n$$

将 T_n 和 T_{2n} 代入有

$$\begin{aligned}\overline{T} = &\frac{4}{3}\left\{\frac{b-a}{4n}\left[f(a) + 2\sum_{i=1}^{2n-1} f\left(a + i\frac{b-a}{2n}\right) + f(b)\right]\right\} - \\ &\frac{1}{3}\left\{\frac{b-a}{2n}\left[f(a) + 2\sum_{i=1}^{n-1} f\left(a + i\frac{b-a}{n}\right) + f(b)\right]\right\} = \\ &\frac{b-a}{3\times(2n)}\left\{\left[f(a) + 4\sum_{i=0}^{n-1} f\left(a + (2i+1)\frac{b-a}{2n}\right) + 2\sum_{i=1}^{n-1} f\left(a + 2i\frac{b-a}{2n}\right) + f(b)\right]\right\} = S_n\end{aligned}$$

所以

$$S_n = \frac{4}{3}T_{2n} - \frac{1}{3}T_n \tag{3.33}$$

这是比复化梯形公式精度高的辛普森公式。也就是说，用复化梯形公式加倍划分区间前后的两次积分的近似值，可以按照式(3.33) 线性组合出精度更高的复化辛普森积分近似值，从而加速了逼近效果，故把公式(3.33) 称为梯形加速公式。

同样能够证明抛物线加速公式

$$C_n = \frac{16}{15}S_{2n} - \frac{1}{15}S_n \tag{3.34}$$

和龙贝格公式

$$R_n = \frac{64}{63}C_{2n} - \frac{1}{63}C_n \tag{3.35}$$

由此可得，用龙贝格公式计算积分的步骤见表 3.4。

表 3.4

		区间等分数	T_{2^k}	$S_{2^{k-1}}$	$C_{2^{k-2}}$	$R_{2^{k-3}}$
计算公式		$n=2^k$	式(3.18)	式(3.33)	式(3.34)	式(3.35)
k	0	1	(1)T_1			
	1	2	(2)T_2	(3)S_1		
	2	4	(4)T_4	(5)S_2	(6)C_1	
	3	8	(7)T_8	(8)S_4	(9)C_2	(10)R_1
	4	16	(11)T_{16}	(12)S_8	(13)C_4	(14)R_2
	5	32	(15)T_{32}	(16)S_{16}	(17)C_8	(18)R_4
	⋮	⋮	⋮	⋮	⋮	⋮

表 3.4 中带括号的数字表示计算次序。一直计算到前、后两个 R_n 值之差的绝对值 $|R_{2n}-R_n|$ 不超过给定的误差为止，并将 R_{2n} 作为积分的近似值。

例 3.5 用龙贝格方法计算 $\pi=\int_0^1 \frac{4}{1+x^2}\mathrm{d}x$ 的近似值，要求结果误差的绝对值 $\varepsilon \leqslant 10^{-5}$。

解 根据式(3.18) 和式(3.32) ～ 式(3.35)，按照表 3.4 的计算次序进行计算，计算结果列入表 3.5 中。

表 3.5

k	区间等分数 ($n=2^k$)	T_{2^k}	$S_{2^{k-1}}$	$C_{2^{k-2}}$	$R_{2^{k-3}}$
0	1	3.000 00			
1	2	3.100 00	3.133 33		
2	4	3.131 18	3.141 57	3.142 12	
3	8	3.138 99	3.141 59	3.141 59	3.141 58
4	16	3.140 94	3.141 59	3.141 59	3.141 59

由于 $|R_2-R_1|=0.000\,01=10^{-5}$，故停止计算，并以 $R_2=3.141\,59$ 作为积分近似值。

与例 3.4 相比，要得到符合精度要求的结果，用龙贝格方法只需计算到 T_{16}，而直接用复化梯形公式必须计算到 T_{256}。显而易见，龙贝格方法收剑速度很快，并且运算量大幅减小。

龙贝格公式计算数值积分的加速过程可以归纳如下：

(1) 计算初值：

$$T_0^{(0)}=\frac{b-a}{2}[f(a)+f(b)]$$

(2) 将区间 $[a,b]$ 二等分，利用复化梯形公式计算：

$$T_0^{(1)}=\frac{1}{2}T_0^{(0)}+\frac{b-a}{2}f\left(a+\frac{1}{2}(b-a)\right)$$

根据式(3.33) 计算：

$$T_1^{(0)} = \frac{4T_0^{(1)} - T_0^{(0)}}{4-1}$$

(3) 再将区间$[a,b]$四等分,利用复化梯形公式计算:

$$T_0^{(2)} = \frac{1}{2}T_0^{(1)} + \frac{b-a}{2^2}\sum_{i=1}^{2} f\left(a + \frac{2i-1}{2^2}(b-a)\right)$$

再根据式(3.33)计算:

$$T_1^{(1)} = \frac{4T_0^{(2)} - T_0^{(1)}}{4-1}$$

(4) 应用式(3.34)计算:

$$T_2^{(0)} = \frac{4^2 T_1^{(1)} - T_1^{(0)}}{4^2-1}$$

(5) 依据公式

$$\left.\begin{aligned} T_0^{(k)} &= \frac{1}{2}T_0^{(k-1)} + \frac{b-a}{2^k}\sum_{i=1}^{2^{k-1}} f\left(a + \frac{2i-1}{2^k}(b-a)\right), & k=1,2,\cdots \\ T_m^{(k)} &= \frac{4^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{4^m-1}, & m=1,2,\cdots \end{aligned}\right\} \tag{3.36}$$

重复上述过程,可得逼近值

$$\begin{array}{cccccccc}
T_0^{(0)} & & & & & & & \\
\downarrow & \searrow & & & & & & \\
T_0^{(1)} & \rightarrow & T_1^{(0)} & & & & & \\
\downarrow & \searrow & \downarrow & \searrow & & & & \\
T_0^{(2)} & \rightarrow & T_1^{(1)} & \rightarrow & T_2^{(0)} & & & \\
\downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & & \\
T_0^{(3)} & \rightarrow & T_1^{(2)} & \rightarrow & T_2^{(1)} & \rightarrow & T_3^{(0)} & \\
\vdots & & \ddots & & \ddots & & \ddots &
\end{array}$$

(6) 精度控制。设预定的数值计算精度为ε,若$|T_m^{(0)} - T_{m-1}^{(0)}| < \varepsilon$,则停止计算,取$T_m^{(0)}$为积分的近似值,否则转入(5)继续计算,直至满足精度要求。

程序(3.6)　用龙贝格方法计算积分的MATLAB程序。

程序任务:用龙贝格方法计算给定函数在区间上的积分。

```
function R = rombergi (funame, a, b, ep)
% 输入:funame—— 被积函数
%        a—— 积分区间下限
%        b—— 积分区间上限
%        ep—— 计算精度(默认值为 1e－5)
% 输出:R—— 数值积分计算结果
if nargin < 4 ep = 1e－5;end
h = b－a;
i = 1; j = 1;
T(i,1) = h/2 * (feval(funame,a) + feval(funame,b));        % 计算 T_1^(1)
```

```
T(i+1,1) = T(i,1)/2 + feval(funame,a + h/2) * h/2;            % 计算 T_1^(2)
T(i+1,j+1) = 4^j * T(i+1,j)/(4^j-1) - T(i,j)/(4^j-1);  % 计算 T_2^(2)
while abs(T(i+1, i+1) - T(i,i)) > ep
    i = i+1; h = h/2;
    T(i+1,1) = T(i,1)/2 + sum(feval(funame,a + h/2:h:b - h/2 + 0.0001 * h)) * h/2;
          % 计算 T_1^(i+1)
    for j = 1:i
        T(i+1,j+1) = 4^j * T(i+1,j)/(4^j-1) - T(i,j)/(4^j-1);   % 计算 T_(j+1)^(i+1)
    end
end
R = T(i+1,j+1);
```

在 MATLAB 命令窗口输入：

```
>> R = rombergi (inline('1 + exp(-x). * sin(4. * x)','x'), 0,1,0.5e-12)
```

% 调用程序(3.6) 计算积分 $\int_0^1 [1+e^{-x}\sin(4x)]\,dx$

输出结果：

R = 1.30825060464267

3.6 数值微分

用函数在一些离散点上的值来推算函数在某点处导数近似值的问题称为数值微分。

3.6.1 差商型数值微分公式

定义(3.5) 函数 $f(x)$ 在 x_0 点处的微分(导数)是

$$f'(x_0)=\lim_{\Delta x\to 0}\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}$$

因此,可以用 $|\Delta x|$ 很小时的比值 $\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}$ (即函数的一阶差商) 作为 $f'(x_0)$ 的近似值。即

$$f'(x_0)\approx\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x} \tag{3.37}$$

这样建立的数值微分公式称为差商型数值微分公式。

通常分为以下几种：

(1) 用向前差商做近似：

$$f'(x_0)\approx\frac{1}{h}[f(x_0+h)-f(x_0)] \tag{3.38}$$

(2) 用向后差商做近似：

$$f'(x_0)\approx\frac{1}{h}[f(x_0)-f(x_0-h)] \tag{3.39}$$

(3) 用中心差商做近似：

$$f'(x_0) \approx \frac{1}{2h}[f(x_0+h)-f(x_0-h)] \tag{3.40}$$

称第(3)种方法为中心方法,相应的计算公式称为中心公式。

下面分析三种差商型数值微分公式的误差。将 $f(x_0 \pm h)$ 在 $x=x_0$ 处做泰勒级数展开,可得

$$f(x_0 \pm h)=f(x_0) \pm f'(x_0)h+\frac{h^2}{2!}f''(x_0) \pm \frac{h^3}{3!}f'''(x_0)+\frac{h^4}{4!}f^{(4)}(x_0) \pm \cdots$$

于是

$$\frac{f(x_0 \pm h)-f(x_0)}{\pm h}=f'(x_0) \pm \frac{h}{2!}f''(x_0)+\frac{h^2}{3!}f'''(x_0)+\cdots$$

所以,向前和向后差商近似公式的截断误差都是 $o(h)$,而中心公式(3.40)的截断误差是 $-h^2f^{(3)}(c)/6=o(h^2)$,这里 $c=c(x_0) \in [a,b]$。

程序(3.7)　用极限方法计算微分近似值的 MATLAB 程序。

程序任务:用极限方法计算给定函数在固定点处微分的近似值。首先生成 $f'(x)$ 近似值的序列

$$f'(x) \approx D_k=\frac{f(x+10^{-k}h)-f(x-10^{-k}h)}{2(10^{-k}h)}, \quad k=0,1,2,\cdots,n$$

当 $|D_n-D_{n-1}| \geqslant |D_{n-1}-D_{n-2}|$ 时,停止计算;当 $|D_n-D_{n-1}|<$ 允许的误差时,停止计算并取 $f'(x) \approx D_n$。

```
function [L,n]=difflim (f, x, ep)
% 输入:f—— 被微分的函数
%      x—— 所求微分点的自变量值
%      ep—— 计算结果的相对误差限
% 输出:L = [H' D' E']。其中 H 是步长向量,D 是微分近似值向量,E 是偏差 |Dn - Dn-1| 组成的向量
%      n—— 最佳近似值的排序
maxl = 15; h = 1;                                             %h 初值为 1,最小值为 10^-maxl
H(1) = h;
D(1) = (feval(f,x+h) - feval(f,x-h))/(2 * h);                 % 计算 D0
E(1) = 0; R(1) = 0;
for n = 1:2
    h = h/10;
    H(n+1) = h;
    D(n+1) = (feval(f,x+h) - feval(f,x-h))/(2 * h);           % 计算 Dn
    E(n+1) = abs(D(n+1) - D(n));                              %Dn 与 Dn-1 的偏差
    R(n+1) = 2 * E(n+1)/(abs(D(n+1)) + abs(D(n)) + eps);      % 相对偏差
end
n = 2;
while ((E(n) > E(n+1)) & (R(n) > ep)) & n < maxl             % 满足条件时进入循环
    h = h/10;
    H(n+2) = h;
    D(n+2) = (feval(f,x+h) - feval(f,x-h))/(2 * h);           % 计算 Dn+1
    E(n+2) = abs(D(n+2) - D(n+1));                            %Dn+1 与 Dn 的偏差
```

```
    R(n+2) = 2 * E(n+2)/ (abs(D(n+2))+abs(D(n+1))+eps);   % 相对偏差
    n = n+1;
end
n = length(D) - 1;                                          % 最佳近似值排序
L = [H' D' E'];
```

在 MATLAB 命令窗口输入：

```
>> [L,n]=difflim (inline('sqrt(x)'),1,1e-5)
```

% 用程序(3.7) 计算 $f(x)=\sqrt{x}$ 在 $x=1$ 处的微分值

输出结果：

```
L =1.000 0     0.707 1     0
   0.100 0     0.500 6     0.206 5
   0.010 0     0.500 0     0.000 6
   0.001 0     0.500 0     0.000 0
   0.000 1     0.500 0     0.000 0
   0.000 0     0.500 0     0.000 0
n =5
```

即 $f'(1)\approx 0.5000$，并且取 $f'(1)\approx D_5$。

3.6.2 理查森外推法

设 $f_k=f(x_k)=f(x_0+kh)$，并且用 $D_0(h)$ 和 $D_0(2h)$ 分别表示以 h 和 $2h$ 为步长，根据式(3.40) 得到的 $f'(x_0)$ 的近似值，则有

$$f'(x_0)\approx D_0(h)+Ch^2 \tag{3.41}$$

$$f'(x_0)\approx D_0(2h)+4Ch^2 \tag{3.42}$$

式(3.41) 乘以 4 减去式(3.42)，可得

$$f'(x_0)\approx \frac{4D_0(h)-D_0(2h)}{3}=\frac{-f_2+8f_1-8f_{-1}+f_{-2}}{12h} \tag{3.43}$$

上式的截断误差是 $h^4f^{(5)}(c_1)/30=o(h^4)$，其中 $c_1=c_1(x_0)\in[a,b]$。

用 $D_1(h)$ 和 $D_1(2h)$ 分别表示用步长 h 和 $2h$，根据式(3.43) 得到的精度为 $o(h^4)$ 的 $f'(x_0)$ 的近似值，它们分别表示为

$$f'(x_0)=\frac{-f_2+8f_1-8f_{-1}+f_{-2}}{12h}+\frac{h^4f^{(5)}(c_1)}{30}\approx D_1(h)+C'h^4$$

$$f'(x_0)=\frac{-f_4+8f_2-8f_{-2}+f_{-4}}{24h}+\frac{16h^4f^{(5)}(c_2)}{30}\approx D_1(2h)+C''h^4$$

若 $f^{(5)}(x)$ 只取正值或负值，而且变化缓慢，则可假设 $f^{(5)}(c_1)\approx f^{(5)}(c_2)$，这时就能从以上两式推导出

$$f'(x_0)\approx \frac{16D_1(h)-D_1(2h)}{15} \tag{3.44}$$

上式的截断误差是 $o(h^6)$。

前面这种从求解 $f'(x_0)$ 的低阶公式推导出高阶公式的方法称为外推法。一般地有理查

森(Richardson)外推法：

设 $f'(x_0)$ 的两个精度为 $o(h^{2k})$ 的近似值分别为 $D_{k-1}(h)$ 和 $D_{k-1}(2h)$，它们分别满足

$$f'(x_0)=D_{k-1}(h)+c_1h^{2k}+c_2h^{2k+2}+\cdots$$
$$f'(x_0)=D_{k-1}(2h)+4^kc_1h^{2k}+4^{k+1}c_2h^{2k+2}+\cdots$$

从中可以得到改进的近似表达式

$$f'(x_0)=D_k(h)+o(h^{2k+2})=\frac{4^kD_{k-1}(h)-D_{k-1}(2h)}{4^k-1}+o(h^{2k+2}) \tag{3.45}$$

例 3.6　设 $f(x)=\cos x$，利用式(3.40)和式(3.43)，步长分别为 $h=0.1,0.01,0.001,0.000\,1$，计算 $f'(0.8)$ 的近似值，并与其真实值 $f'(0.8)=-\sin 0.8$ 进行比较。

解　取 $h=0.01$，根据式(3.40)，可得

$$f'(0.8)\approx\frac{f(0.81)-f(0.79)}{0.02}\approx\frac{0.689\,498\,433-0.703\,845\,316}{0.02}\approx-0.717\,344\,150$$

根据式(3.43)，可得

$$f'(0.8)\approx\frac{-f(0.82)+8f(0.81)-8f(0.79)+f(0.78)}{0.12}\approx$$
$$\frac{-0.682\,221\,207+8\times0.689\,498\,433-8\times0.703\,845\,316+0.710\,913\,538}{0.12}\approx$$
$$-0.717\,356\,108$$

根据两式所计算近似值的误差分别为 $-0.000\,011\,941$ 和 $0.000\,000\,017$。其他近似值列于表 3.6 中。

表　3.6

步长	式(3.40)的值	式(3.40)的误差	式(3.43)的值	式(3.43)的误差
0.1	−0.716 161 095	−0.001 194 996	−0.717 353 703	−0.000 002 389
0.01	−0.717 344 150	−0.000 011 941	−0.717 356 108	0.000 000 017
0.001	−0.717 356 000	−0.000 000 091	−0.717 356 167	0.000 000 076
0.000 1	−0.717 360 000	−0.000 003 909	−0.717 360 833	0.000 004 742

程序(3.8)　用理查森外推法计算微分近似值的 MATLAB 程序。

程序任务：用理查森外推法计算给定函数在固定点处微分的近似值。首先构造 $f'(x)$ 近似值的表 $D(j,k)$ $(k\leqslant j)$，并将 $f'(x)\approx D(n,n)$ 作为最终答案。近似值 $D(j,k)$ 存放在下三角矩阵中。第一列是

$$D(j,0)=\frac{f(x+2^{-j}h)-f(x-2^{-j}h)}{2^{-j+1}h}$$

第 j 行的元素为

$$D(j,k)=D(j,k-1)+\frac{D(j,k-1)-D(j-1,k-1)}{4^k-1}\quad(1\leqslant k\leqslant j)$$

```
function [D,err,relerr,n]=diffRich (f, x, delta, ep)
% 输入:f—— 被微分的函数
%       x—— 所求微分点的自变量值
%       delta—— 计算结果的误差限
```

```
%       ep—— 计算结果的相对误差限
% 输出:D—— 微分近似值矩阵
%       err—— 误差界
%       relerr—— 相对误差界
%       n—— 最佳近似值的排序
err = 1; relerr = 1; h = 1;
j = 1;
D(1,1) = (feval(f,x + h) - feval(f,x - h))/(2 * h);              % 计算 D(0,0)
while relerr > ep & err > delta & j < 12                           % 满足条件时进入循环
    h = h/2;
    D(j + 1,1) = (feval(f,x + h) - feval(f,x - h))/(2 * h);        % 计算 D(j,0)
    for k = 1:j
        D(j + 1,k + 1) = D(j + 1,k) + ( D(j + 1,k) - D(j,k))/((4^k) - 1);   % 计算 D(j,k)
    end
    err = abs(D(j + 1,j + 1) - D(j,j));                            % 误差中间值
    relerr = 2 * err/(abs(D(j + 1,j + 1)) + abs( D(j,j)) + eps)    % 相对误差中间值
    j = j + 1;
end
[n,n] = size(D);                                                   % 提取最佳近似值的排序
```

在 MATLAB 命令窗口输入：

```
>> [D,err,relerr,n]=diffRich (inline('sqrt(x)'),1,1e-3,1e-5)
```

% 用程序(3.8) 计算 $f(x)=\sqrt{x}$ 在 $x=1$ 处的微分值

输出结果：

```
D =0.707 1    0         0         0         0
   0.517 6    0.454 5   0         0         0
   0.504 0    0.499 5   0.502 5   0         0
   0.501 0    0.500 0   0.500 0   0.500 0   0
   0.500 2    0.500 0   0.500 0   0.500 0   0.500 0
err =3.4331e-005
relerr =6.8664e-005
n =5
```

即 $f'(1)\approx 0.500\,000\,120$，并且取 $f'(1)\approx D(5,5)$。

3.6.3 插值型数值微分公式

定义(3.6) 当函数 $f(x)$ 以表格形式 $y_i=f(x_i)$ $(i=0,1,\cdots,n)$ 给出时，若用插值多项式 $P_n(x)$ 作为函数 $f(x)$ 的近似函数 $f(x)\approx P_n(x)$，则可把 $P_n(x)$ 的导数 $P'_n(x)$ 作为 $f'(x)$ 的近似值，这样建立的数值微分公式

$$f'(x)\approx P'_n(x) \tag{3.46}$$

称为插值型数值微分公式。

插值型数值微分公式的截断误差从插值多项式的余项得到。由于

$$f(x)=P_n(x)+R_n(x)=P_n(x)+\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)\quad (a<\xi<b)$$

两边求导得

$$f'(x)=P'_n(x)+R'_n(x)$$

其中

$$R'_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega'_{n+1}(x)+\frac{\omega_{n+1}(x)}{(n+1)!}\frac{\mathrm{d}}{\mathrm{d}x}(f^{(n+1)}(\xi))$$

因为 ξ 是 x 的未知函数，上式的最后一项无法计算。但是，如果求节点 x_i 处的导数值，则截断误差为

$$R'_n(x_i)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega'_{n+1}(x_i)\tag{3.47}$$

用插值多项式 $P_n(x)$ 作为 $f(x)$ 的近似函数，也可以建立高阶数值微分公式：

$$f^{(k)}(x)\approx P_n^{(k)}(x)\quad (k=1,2,3,\cdots)\tag{3.48}$$

下面给出几个常用的计算一阶微分的插值型数值微分公式：

1. 两点公式

若 $f'(x)$ 连续，$f''(x)$ 存在，且已知点 $(x_0,f(x_0))$ 和 $(x_1,f(x_1))$，则线性插值多项式 $P_1(x)$ 为

$$P_1(x)=\frac{x-x_1}{x_0-x_1}f(x_0)+\frac{x-x_0}{x_1-x_0}f(x_1)$$

对 $P_1(x)$ 求导，得出计算一阶导数的两点公式

$$f'(x_0)\approx\frac{1}{h}[f(x_1)-f(x_0)],\quad f'(x_1)\approx\frac{1}{h}[f(x_1)-f(x_0)]\tag{3.49}$$

式中，$h=x_1-x_0$。公式的截断误差分别是

$$R_1(x_0)=-\frac{h}{2}f''(\xi),\quad R_1(x_1)=\frac{h}{2}f''(\xi)\tag{3.50}$$

2. 三点公式

若 $f''(x)$ 连续，$f'''(x)$ 存在，且已知点 $(x_i,f(x_i))\ (i=0,1,2)$，则建立二次插值多项式 $P_2(x)$ 为

$$P_2(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0)+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1)+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2)$$

对 $P_2(x)$ 求导，得出计算一阶导数的三点公式

$$\left.\begin{aligned}f'(x_0)&\approx\frac{1}{2h}[-3f(x_0)+4f(x_1)-f(x_2)]\\ f'(x_1)&\approx\frac{1}{2h}[f(x_2)-f(x_0)]\\ f'(x_2)&\approx\frac{1}{2h}[f(x_0)-4f(x_1)+3f(x_2)]\end{aligned}\right\}\tag{3.51}$$

式中，$h=x_1-x_0=x_2-x_1$。公式的截断误差分别是

$$R_2(x_0)=\frac{h^2}{3}f'''(\xi),\quad R_2(x_1)=-\frac{h^2}{6}f'''(\xi),\quad R_2(x_2)=\frac{h^2}{3}f'''(\xi)\tag{3.52}$$

程序(3.9) 基于牛顿插值多项式计算微分近似值的 MATLAB 程序。

程序任务：构造牛顿插值多项式，并计算给定点处微分的近似值。首先建立 N 次牛顿插值多项式

$$N(x)=a_0+a_1(x-x_0)+a_2(x-x_0)(x-x_1)+\cdots+a_N(x-x_0)\cdots(x-x_{N-1})$$

再将 $f'(x_0)\approx N'(x_0)$ 作为最终结果。若在 x_d 处使用这个方法，可以通过重新排列点的顺序 $\{x_k,x_d,\cdots,x_{k-1},x_{k+1},\cdots x_N\}$ 来计算 $f'(x_d)\approx P'(x_d)$。

```
function [A,dif]=diffnewton(x,y,xd)
% 输入:x—— 插值节点向量
%     y—— 插值节点处的函数值向量
%     xd—— 计算微分处的坐标
% 输出:A—— 牛顿插值多项式的系数向量
%     dif—— 微分近似值
A = y;
N = length(x);
for j = 2:N                          % 计算牛顿插值多项式的系数
    for k = N:-1:j
        A(k) = (A(k) - A(k-1))/(x(k) - x(k-j+1));
    end
end
dif = A(2);
prod = 1;
n1 = length(A) - 1;
for k = 2:n1                         % 计算微分
    prod = prod * (xd - x(k));
    dif = dif + prod * A(k+1);
end
```

在 MATLAB 命令窗口输入：

```
>> x=0.99:0.001:1.01;                % 建立插值节点向量
>> y=sqrt(x);                        % 建立插值节点处的函数值向量
>> [A,dif]=diffnewton(x,y,1)         %调用程序(3.9)计算 xd=1 处微分的近似值
```

输出结果：

```
dif = 0.50125628933800
```

3.7 物理学中的应用举例

3.7.1 均匀带电直线段与均匀带电圆环的电场

静电场中某点电场强度的大小和方向与单位正电荷在该点受力的大小和方向分别相同。即

$$\boldsymbol{E}=\frac{\boldsymbol{F}}{q_0} \tag{3.53}$$

式中，$\boldsymbol{F}$ 是试验电荷 q_0 受到的电场力。电场强度的单位是 N/C。而静电场中某点的电势在数值上等于把单位正电荷从该点沿任意路径移动到电势零参考点的过程中静电力所做的功。表示为

$$u_p=\frac{W_p}{q_0} \tag{3.54}$$

式中，W_p 是把试验电荷 q_0 从 p 点移动到电势零参考点"0"过程中静电力所做的功。电势的单位是 V。空间 p 点电势与电场强度的关系是

$$u_p=\int_p^{\text{“0”}}\boldsymbol{E}\cdot\mathrm{d}\boldsymbol{l} \tag{3.55}$$

和

$$\boldsymbol{E}=-\operatorname{grad}(u_p)=-\left(\frac{\partial u_p}{\partial x}\boldsymbol{i}+\frac{\partial u_p}{\partial y}\boldsymbol{j}+\frac{\partial u_p}{\partial z}\boldsymbol{k}\right) \tag{3.56}$$

其中，$\operatorname{grad}(u_p)$ 表示函数 u_p 的梯度，$\boldsymbol{i},\boldsymbol{j},\boldsymbol{k}$ 分别是直角坐标系 $Oxyz$ 三个坐标轴方向的单位矢量。

真空中的点电荷 q 产生的电场的电场强度与电势分别为

$$\boldsymbol{E}(\boldsymbol{r})=\frac{q}{4\pi\varepsilon_0 r^2}\boldsymbol{r}_0 \tag{3.57}$$

与

$$u(\boldsymbol{r})=\frac{q}{4\pi\varepsilon_0 r} \tag{3.58}$$

这里，ε_0 是真空的介电常数，$\boldsymbol{r}$ 是空间点相对于点电荷的位置矢量，$r=|\boldsymbol{r}|$，单位矢量 $\boldsymbol{r}_0=\boldsymbol{r}/r$。根据电场强度和电势的叠加原理，带电量为 Q 的带电体所产生电场的电场强度和电势分别为

$$\boldsymbol{E}(\boldsymbol{r})=\frac{1}{4\pi\varepsilon_0}\int_Q\frac{\mathrm{d}q}{r^2}\boldsymbol{r}_0 \tag{3.59}$$

和

$$u(\boldsymbol{r})=\frac{1}{4\pi\varepsilon_0}\int_Q\frac{\mathrm{d}q}{r} \tag{3.60}$$

显然，计算电场强度有两种途径：一是根据已知电荷分布，应用式(3.59)用积分方法计算；二是若已知电势分布，利用式(3.56)用微分方法计算。同样，计算电势也有两种方法：一是根据已知电荷分布，应用式(3.60)用积分方法计算；二是若已知电场强度分布，利用式(3.55)用积分方法计算。

1. 均匀带电直线段的电势与电场强度

设有一均匀带电直线段，长度为 L，带电量为 q。点 P 到直线段的距离为 a，P 与直线段两端连线与 y 轴正方向夹角分别为 θ_1 和 θ_2，如图 3.3 所示。选取点 P 到直线段的垂足为原点建立坐标系，坐标轴如图 3.3 所示。在带电直线段上距原点 y 处取一线元 $\mathrm{d}y$，其上带电量为 $\mathrm{d}q=\lambda\mathrm{d}y$，其中 $\lambda=q/L$ 为电荷线密度。取无穷远处为电势零参考点，设 $\mathrm{d}y$ 到点 P 的距离为 r，则点电

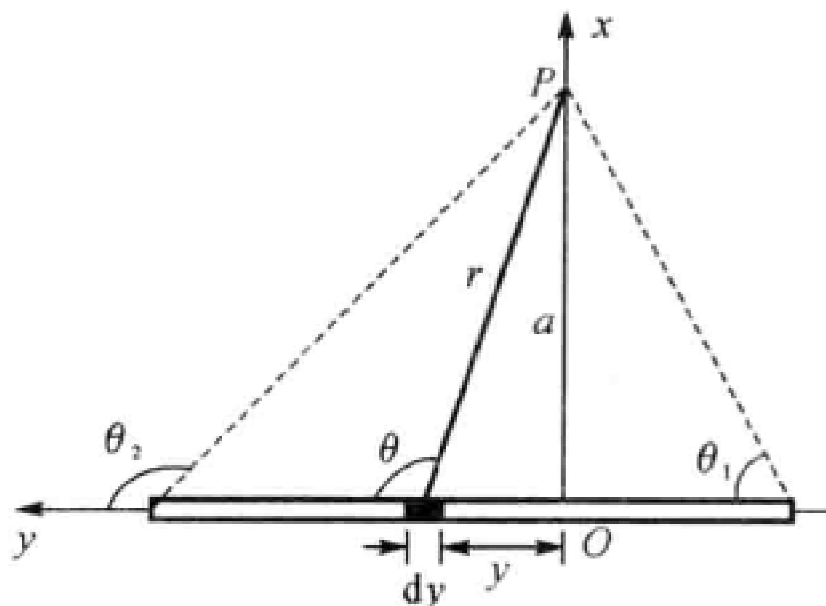


图 3.3

荷 dq 在点 P 产生的电势为

$$du=\frac{\lambda dy}{4\pi\varepsilon_0 r} \tag{3.61}$$

带电直线段在 P 点产生的电势为

$$u=\frac{\lambda}{4\pi\varepsilon_0}\int\frac{dy}{r}=\frac{\lambda}{4\pi\varepsilon_0}\int_{y_1}^{y_2}\frac{dy}{\sqrt{a^2+y^2}}=\frac{\lambda}{4\pi\varepsilon_0}\ln\frac{y_2+\sqrt{a^2+y_2^2}}{y_1+\sqrt{a^2+y_1^2}} \tag{3.62}$$

式中

$$y_1=-a\cot\theta_1,\quad y_2=-a\cot\theta_2 \tag{3.63}$$

为了推导出式(3.62) 更为简洁的表达式，进行积分变量转换。由图 3.3 中的几何关系可知

$$\begin{cases}y=a\tan\left(\theta-\dfrac{\pi}{2}\right)=-a\cot\theta\\ dy=a\csc^2\theta d\theta\\ r=\sqrt{a^2+y^2}=a\csc\theta\end{cases}$$

将以上三式代入式(3.61)，得

$$du=\frac{\lambda}{4\pi\varepsilon_0}\csc\theta d\theta$$

对其积分就是直线段产生的电势，即

$$u=\frac{\lambda}{4\pi\varepsilon_0}\int_{\theta_1}^{\theta_2}\csc\theta d\theta=\frac{\lambda}{4\pi\varepsilon_0}\left[\ln\left(\tan\frac{\theta_2}{2}\right)-\ln\left(\tan\frac{\theta_1}{2}\right)\right] \tag{3.64}$$

带电直线段在点 P 产生的电场强度为[①]

$$E_x=\frac{\lambda}{4\pi\varepsilon_0 a}(\cos\theta_1-\cos\theta_2) \tag{3.65}$$

$$E_y=\frac{\lambda}{4\pi\varepsilon_0 a}(\sin\theta_2-\sin\theta_1) \tag{3.66}$$

令

$$\alpha=-\frac{y_1}{L},\quad \beta=\frac{a}{L} \tag{3.67}$$

则式(3.62)、式(3.65) 和式(3.66) 分别表示为

$$u=\frac{q}{4\pi\varepsilon_0 L}\int_{-\alpha}^{1-\alpha}\frac{dz}{\sqrt{z^2+\beta^2}}=\frac{q}{4\pi\varepsilon_0 L}\ln\frac{\sqrt{(1-\alpha)^2+\beta^2}+(1-\alpha)}{\sqrt{\alpha^2+\beta^2}-\alpha} \tag{3.68}$$

$$E_x=\frac{q}{4\pi\varepsilon_0 L^2}\frac{1}{\beta}\left[\frac{(1-\alpha)}{\sqrt{(1-\alpha)^2+\beta^2}}+\frac{\alpha}{\sqrt{\alpha^2+\beta^2}}\right] \tag{3.69}$$

$$E_y=\frac{q}{4\pi\varepsilon_0 L^2}\left[\frac{1}{\sqrt{(1-\alpha)^2+\beta^2}}-\frac{1}{\sqrt{\alpha^2+\beta^2}}\right] \tag{3.70}$$

位于距离直线段右端 $L/4$ 的垂线上的各点(见图 3.4)，有 $\alpha=-y_1/L=1/4$，则式(3.68) 和式(3.69) 分别简化为

$$u=\frac{q}{4\pi\varepsilon_0 L}\int_{-1/4}^{3/4}\frac{dz}{\sqrt{z^2+\beta^2}}=\frac{q}{4\pi\varepsilon_0 L}\ln\frac{\sqrt{(3/4)^2+\beta^2}+3/4}{\sqrt{(1/4)^2+\beta^2}-1/4} \tag{3.71}$$

① 吴百诗. 大学物理基础(下册). 北京:科学出版社,2007.

$$E_x = \frac{q}{4\pi\varepsilon_0 L^2}\frac{1}{\beta}\left[\frac{3/4}{\sqrt{(3/4)^2+\beta^2}}+\frac{1/4}{\sqrt{(1/4)^2+\beta^2}}\right] \tag{3.72}$$

从式(3.71) 计算 E_x 的公式为

$$E_x = -\frac{\mathrm{d}u}{\mathrm{d}a} = -\frac{1}{L}\frac{\mathrm{d}u}{\mathrm{d}\beta} \tag{3.73}$$

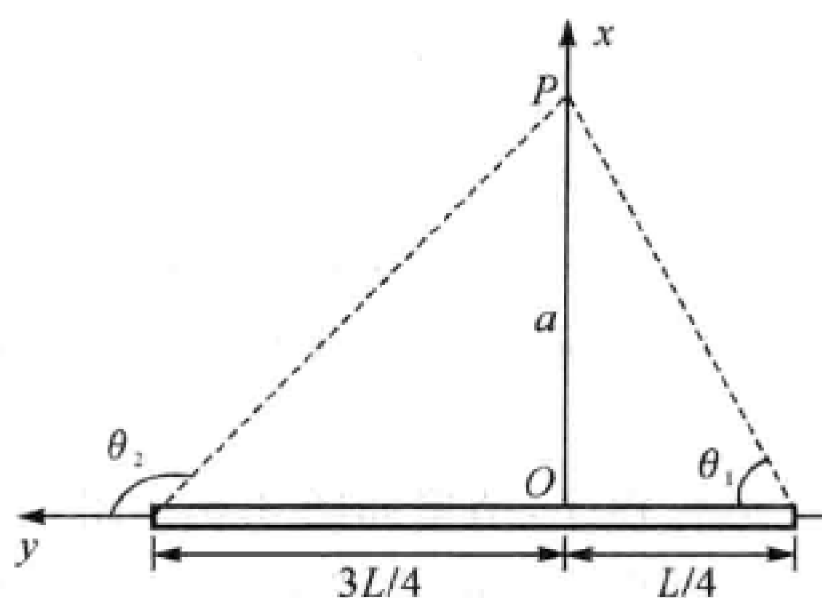


图　3.4

程序(3.10)　分析均匀带电直线段电场中电势与电场强度的 MATLAB 程序。

程序任务：取 $q/4\pi\varepsilon_0=1$ 及 $L=1$。首先用数值积分方法计算 β 取不同值时式(3.71) 第一个等号后的积分值，以便计算电势 $u(\beta)$ 的近似值，并与式(3.71) 第二个等号后解析式的电势计算结果进行比较；然后用数值微分方法计算 β 取不同值时式(3.71) 第二个等号后的电势函数微分的近似值，计算电场强度分量，并与解析式(3.72) 计算结果进行比较。

```
% 以下程序段用数值积分方法计算式(3.71) 第一个等号后的积分近似值，计算电势，并与式(3.71) 第
  二个等号后解析式的电势计算结果进行比较。
global bt
qe = 1; l = 1;          % 为简化计算公式的取值
N = 10;                 %N 是数值计算积分和微分点的数目
bt00 = 1; bt01 = 1; %bt00 和 bt01 分别是计算点序列起点坐标和间隔的 β 值
for k = 1:1:N;
    bt1(k) = bt00 + (k - 1) * bt01;
    bt = bt1(k);
    Iv(k) = (qe/l) * changtrap ('funele1', - 1/4,3/4,1e - 3);
        % 用变步长梯形积分程序(3.5) 计算式(3.71) 第一个等号后的积分，调用了函数文件
          funele1.m
end
bt0 = bt1(1):(bt1(2) - bt1(1))./10:max(bt1);
fv = (qe./l).* log((sqrt(0.75.^2 + bt0.^2) + 0.75)./ (sqrt(0.25.^2 + bt0.^2) - 0.25));
        % 式(3.71) 第二个等号后解析表达式的计算
figure;
plot(bt0,fv,'-k',bt1,Iv,'*k');
        % 绘制电势变化曲线，实线为解析表达式曲线，"*"表示数值积分计算结果
xlabel('\beta'); ylabel('u(\beta)');       % 标注坐标轴
% 以下程序段用数值微分方法计算式(3.71) 式第二个等号后的电势微分的近似值，计算电场强度分
  量，并与解析式(3.72) 计算结果进行比较。
for k = 1:1:N
    [L,n]=difflim ('funele2',bt1(k),1e-2);
        %用计算微分的极限方法程序(3.7)计算式(3.71)第二个等号后电势函数的微分，调用了函数
          文件 funele2.m
    IE(k)=-L(n,2); %计算电场强度分量近似值
end
fE=(qe./l.^2).*(0.75./sqrt(0.75.^2+bt0.^2)+0.25./sqrt(0.25.^2+bt0.^2))./bt0;
```

```
        %电场强度解析式(3.72)的计算结果
figure;
plot(bt0,fE,'-k',bt1,IE,'ok');
        %绘制电场强度分量变化曲线,实线为解析表达式曲线,"o"表示数值微分计算结果
xlabel('\beta');ylabel('E_x(\beta)');    %标注坐标轴
```

运行程序,结果如图 3.5 和图 3.6 所示。图 3.5 是电势变化曲线,实线为式(3.71)第二个等号后解析式的曲线,“∗”表示数值积分计算结果;图 3.6 是电场强度分量变化曲线,实线为解析表达式(3.72)的曲线,“o”表示数值微分计算结果。

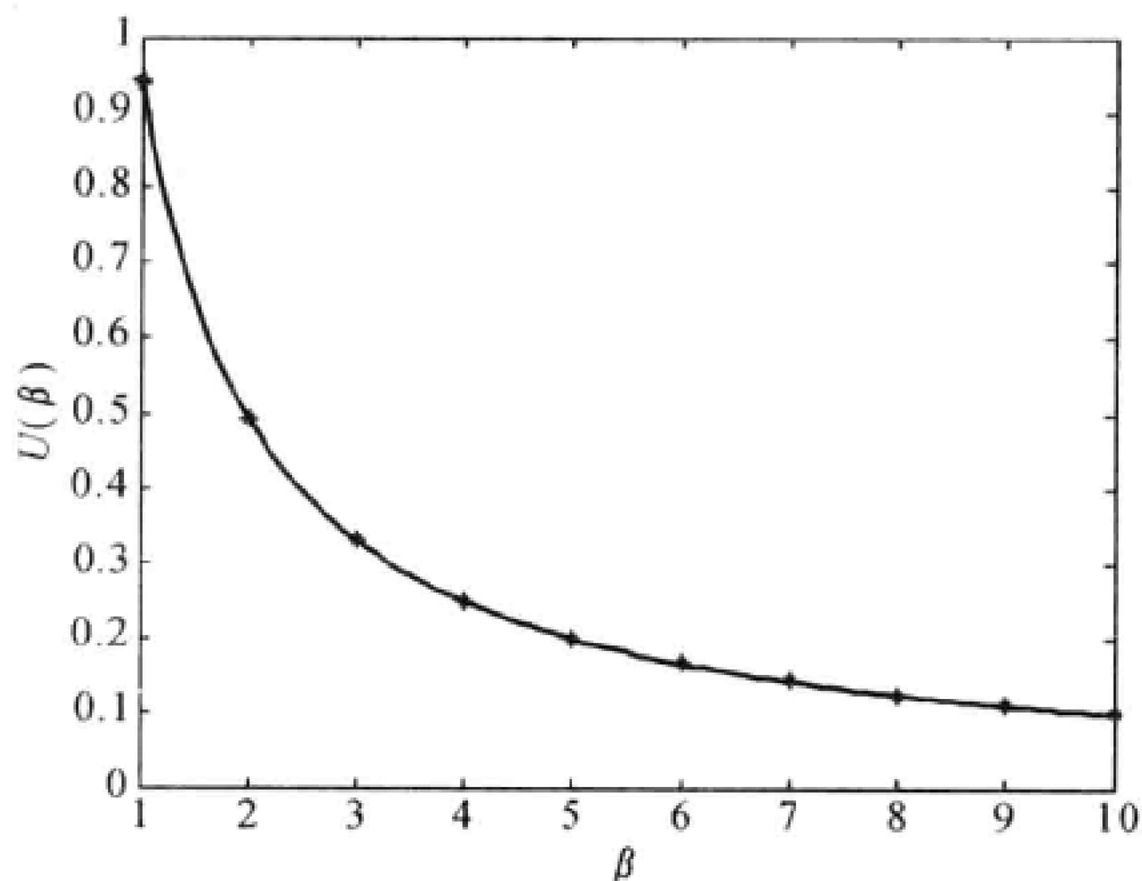


图 3.5

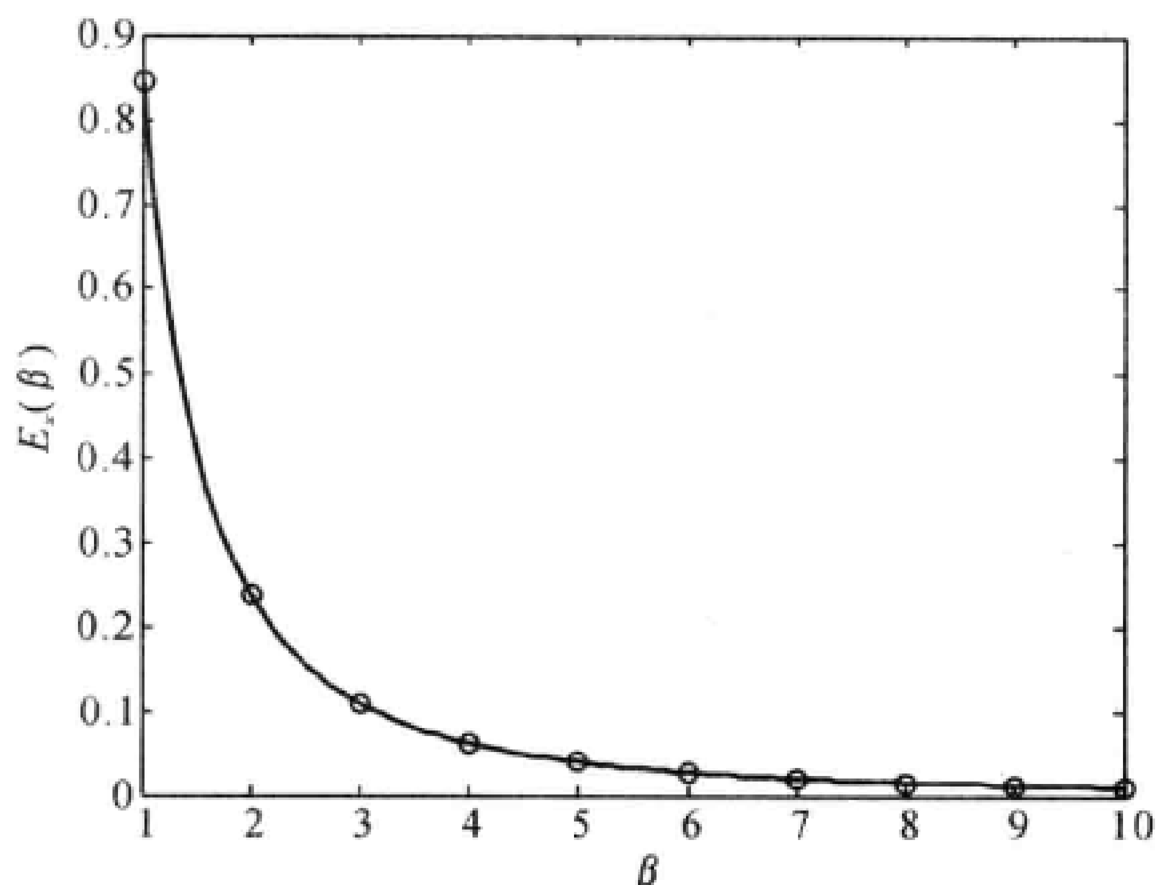


图 3.6

程序(3.10)中调用的函数 funele1 由程序 funele1.m 定义:

```
function y=funele1(x)        %定义程序(3.10)中的被积函数
global bt;
y=1./sqrt(x.^2+bt.^2);
```

函数 funele2 由程序 funele2.m 定义：

```
function y=funele2(x)        % 定义程序(3.10)中的函数
y=log((sqrt(0.75.^2+x.^2)+0.75)./(sqrt(0.25.^2+x.^2)-0.25));
```

2. 均匀带电圆环的电势与电场强度

电荷 q 均匀分布在半径为 R 的圆环上，带电圆环在空间产生电场，为了计算电场在空间的分布，建立如图 3.7 所示的坐标系。根据问题的轴对称性，选取计算的场点 P 在 xOz 坐标面上。圆环上 S 处的点电荷 $dq=(q/2\pi R)R\,d\varphi$ 在场点 P 产生的电势和电场强度分别为[①②]

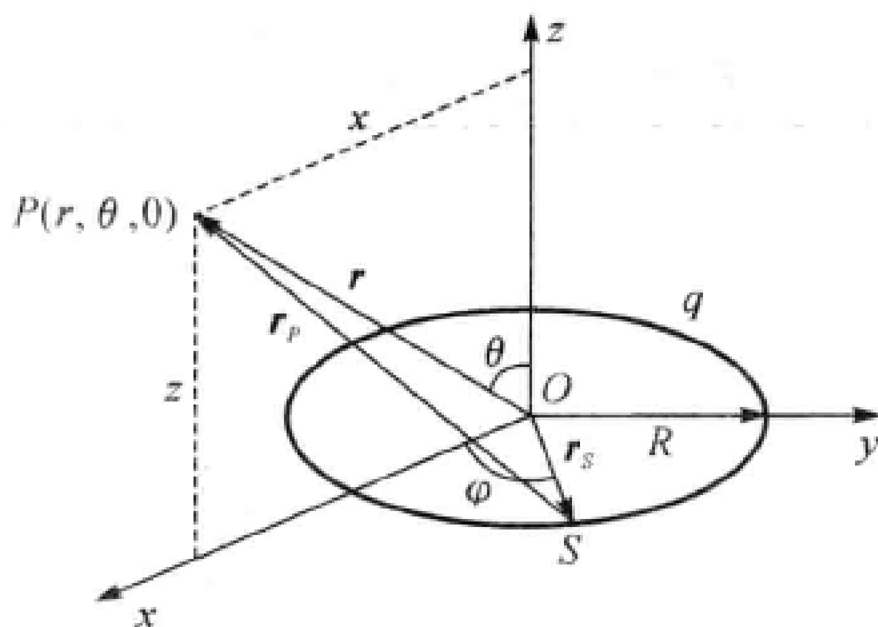


图 3.7

$$du=\frac{dq}{4\pi\varepsilon_0}\frac{1}{|\boldsymbol{r}_P|}=\frac{q}{8\pi^2\varepsilon_0}\frac{d\varphi}{(R^2+r^2-2rR\sin\theta\cos\varphi)^{1/2}} \tag{3.74}$$

$$d\boldsymbol{E}=\frac{dq}{4\pi\varepsilon_0}\frac{\boldsymbol{r}_P}{|\boldsymbol{r}_P|^3}=\frac{q}{8\pi^2\varepsilon_0}\frac{(r\sin\theta-R\cos\varphi)\boldsymbol{i}-R\sin\varphi\boldsymbol{j}+r\cos\theta\boldsymbol{k}}{(R^2+r^2-2rR\sin\theta\cos\varphi)^{3/2}}d\varphi \tag{3.75}$$

对以上两式进行积分，得到圆环上电荷在点 P 产生的电势和电场强度分别为

$$u=\frac{q}{8\pi^2\varepsilon_0}\int_0^{2\pi}\frac{d\varphi}{(R^2+r^2-2rR\sin\theta\cos\varphi)^{1/2}} \tag{3.76}$$

$$\boldsymbol{E}=\frac{q}{8\pi^2\varepsilon_0}\int_0^{2\pi}\frac{(r\sin\theta-R\cos\varphi)\boldsymbol{i}-R\sin\varphi\boldsymbol{j}+r\cos\theta\boldsymbol{k}}{(R^2+r^2-2rR\sin\theta\cos\varphi)^{3/2}}d\varphi \tag{3.77}$$

经过推导，得到电势和电场强度分量的表达式分别为

$$u=\frac{q}{2\pi^2\varepsilon_0}\frac{K(k)}{(R^2+r^2+2rR\sin\theta)^{1/2}} \tag{3.78}$$

$$\left.\begin{aligned}
E_x&=\frac{q}{4\pi^2\varepsilon_0}\frac{1}{r\sin\theta\,(R^2+r^2+2rR\sin\theta)^{1/2}}\left(\frac{2r^2\sin^2\theta-R^2-r^2}{R^2+r^2-2rR\sin\theta}E(k)+K(k)\right)\\
E_y&=0\\
E_z&=\frac{q}{4\pi^2\varepsilon_0}\frac{2r\cos\theta}{(R^2+r^2-2rR\sin\theta)(R^2+r^2+2rR\sin\theta)^{1/2}}E(k)
\end{aligned}\right\} \tag{3.79}$$

式中

$$K(k)=\int_0^{\pi/2}\frac{d\psi}{(1-k^2\sin^2\psi)^{1/2}} \tag{3.80}$$

$$E(k)=\int_0^{\pi/2}(1-k^2\sin^2\psi)^{1/2}d\psi \tag{3.81}$$

分别是第一类全椭圆积分和第二类全椭圆积分。其中

$$k^2=\frac{4rR\sin\theta}{R^2+r^2+2rR\sin\theta} \tag{3.82}$$

应用龙贝格积分程序(3.6)计算所得的第一类、第二类全椭圆积分值列于表 3.7 中，椭圆积分曲线如图 3.8 和图 3.9 所示。

---

① 张之翔. 圆环电荷的电势的几种算法及讨论. 大学物理，2006，25(8)：7－10.

② 张文翔. 均匀带电圆环的电场强度. 大学物理，2012，31(5)：14－16.

表 3.7

k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$K(k)$	1.574 7	1.586 9	1.608 0	1.640 0	1.685 8	1.750 8	1.845 7	1.995 3	2.280 5
$E(k)$	1.566 9	1.555 0	1.534 8	1.505 9	1.467 5	1.418 1	1.355 7	1.276 3	1.171 7

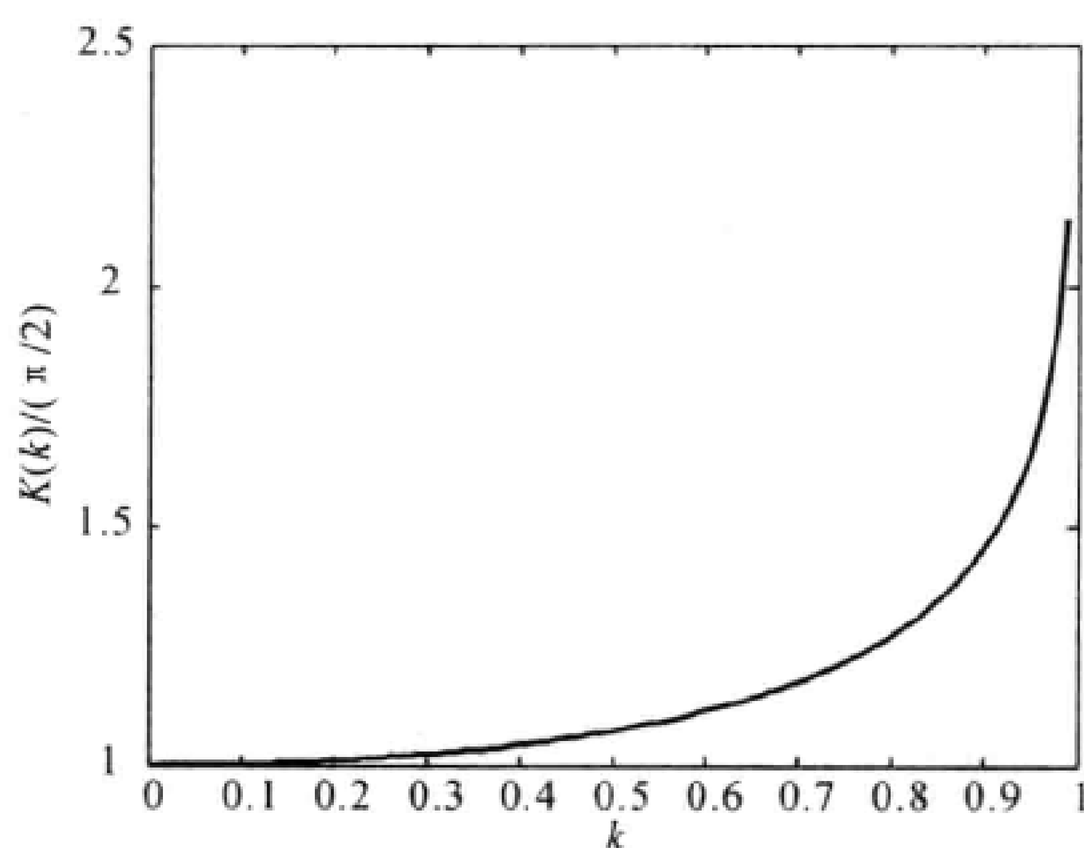


图 3.8

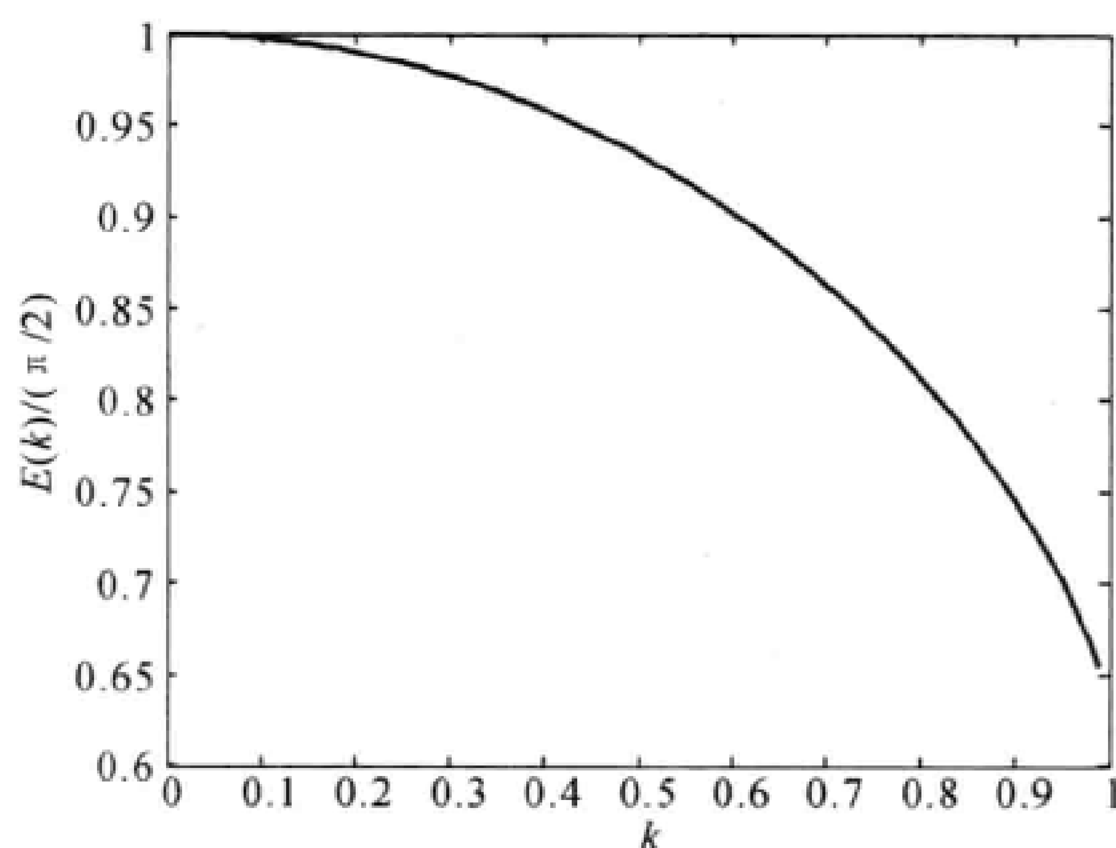


图 3.9

若场点 P 位于圆环的对称轴(即 z 轴)上,即 $\theta=0$ 或 $\theta=\pi$,$\sin\theta=0$,$r=z$,$k=0$,$K(0)=E(0)=\pi/2$,式(3.78)和式(3.79)简化为

$$u(z)=\frac{q}{4\pi\varepsilon_0}\frac{1}{(R^2+z^2)^{1/2}} \tag{3.83}$$

$$E_z(z)=\frac{q}{4\pi\varepsilon_0}\frac{2z}{(R^2+z^2)^{3/2}},\quad E_x(z)=E_y(z)=0 \tag{3.84}$$

若场点 P 位于圆环平面内(即 x 轴)上,则 $\theta=\pi/2$,$\sin\theta=1$,$r=x$,以及

$$k^2=\frac{4xR}{R^2+x^2+2xR}=\frac{4xR}{(R+x)^2}=\frac{4x/R}{(1+x/R)^2} \tag{3.85}$$

式(3.78)简化为

$$u(x)=\frac{q}{2\pi^2\varepsilon_0}\frac{K(k)}{|R+x|}=\frac{q}{2\pi^2\varepsilon_0 R}\frac{K(k)}{|1+x/R|}=\frac{q}{4\pi\varepsilon_0 R}\frac{2}{\pi}\frac{K(k)}{|1+x/R|}=\frac{2u(O)}{\pi}\frac{K(k)}{|1+x/R|} \tag{3.86}$$

其中,$u(O)=q/(4\pi\varepsilon_0 R)$,是圆环中心点 O 的电势。而式(3.79)简化为

$$\left.\begin{aligned}E_x&=\frac{q}{4\pi^2\varepsilon_0 R^2}\frac{1}{(x/R)(1+x/R)}\left(K(k)-\frac{1+x/R}{1-x/R}E(k)\right)=\\&E_0\frac{1}{(x/R)}\left(\frac{K(k)}{1+x/R}-\frac{E(k)}{1-x/R}\right)\\E_y(x)&=E_z(x)=0\end{aligned}\right\} \tag{3.87}$$

其中,$E_0=q/(4\pi^2\varepsilon_0 R^2)$。

程序(3.11)　绘制 $u(x)/u(O)-x/R$ 和 $E(x)/E_0-x/R$ 曲线的 MATLAB 程序。

程序任务:利用龙贝格积分方法计算全椭圆积分,并根据式(3.86)和式(3.87)绘制 $u(x)/u(O)-x/R$ 和 $E(x)/E_0-x/R$ 曲线。

```
xR1 = 0.0:0.05:0.92;                  %xR = x/R   (x < R)
xR2 = 1.05:0.05:5;                    %xR = x/R   (x > R)
k1 = sqrt(4. * xR1)./(1 + xR1);       % 公式(3.85)
k2 = sqrt(4. * xR2)./(1 + xR2);       % 公式(3.85)
global cK cE;
n1 = length(k1);
for I = 1:1:n1
    cK = k1(I); cE = k1(I);
    K1(I) = rombergi('funK',0,pi./2,1e-6);
        % 用龙贝格积分方法程序(3.6)计算式(3.80)定义的第一类全椭圆积分 K(k),调用了函数文
          件 funK.m
    E1(I) = rombergi('funE',0,pi./2,1e-6);
        % 用龙贝格积分方法程序(3.6)计算式(3.81)定义的第二类全椭圆积分 E(k),调用了函数文
          件 funE.m
end
n2 = length(k2);
for I = 1:1:n2
    cK = k2(I); cE = k2(I);
    K2(I) = rombergi('funK',0,pi./2,1e-6);
        % 用龙贝格积分方法程序(3.6)计算式(3.80)定义的第一类全椭圆积分 K(k),调用了函数文
          件 funK.m
    E2(I) = rombergi('funE',0,pi./2,1e-6);
        % 用龙贝格积分方法程序(3.6)计算式(3.81)定义的第二类全椭圆积分 E(k),调用了函数文
          件 funE.m
end
IU1 = 2. * K1./(pi. * (1 + xR1));            % 用公式(3.86)计算 u(x)/u(O)
IU2 = 2. * K2./(pi. * (1 + xR2));            % 用公式(3.86)计算 u(x)/u(O)
IE1 = (1./xR1). * (K1./(1 + xR1) - E1./(1 - xR1));      % 用公式(3.87)计算 E(x)/E0
```

```
IE2 = (1./xR2).*(K2./(1 + xR2) - E2./(1 - xR2));       % 用公式(3.87)计算 E(x)/E0
figure
plot(xR1,IU1,'k',xR2,IU2,'k');       % 绘制 u(x)/u(O) - x/R 曲线
axis([0 5 0 2.0]);
xlabel('x/R');ylabel('U/U(O)');
figure;
plot(xR1,IE1,'k',xR2,IE2,'k');       % 绘制 E(x)/E0 - x/R 曲线
axis([0 5 -10 15]);
xlabel('x/R');ylabel('E/E_0');
```

程序运行结果如图 3.10 和图 3.11 所示。

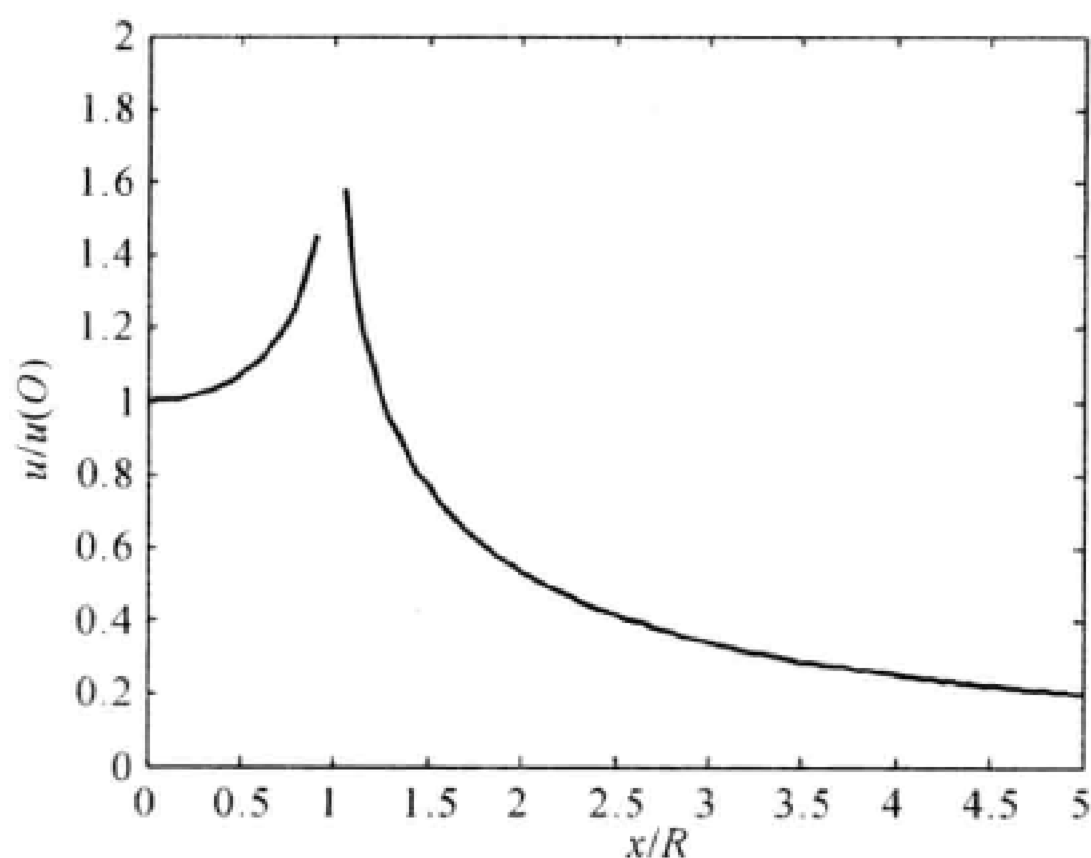


图 3.10

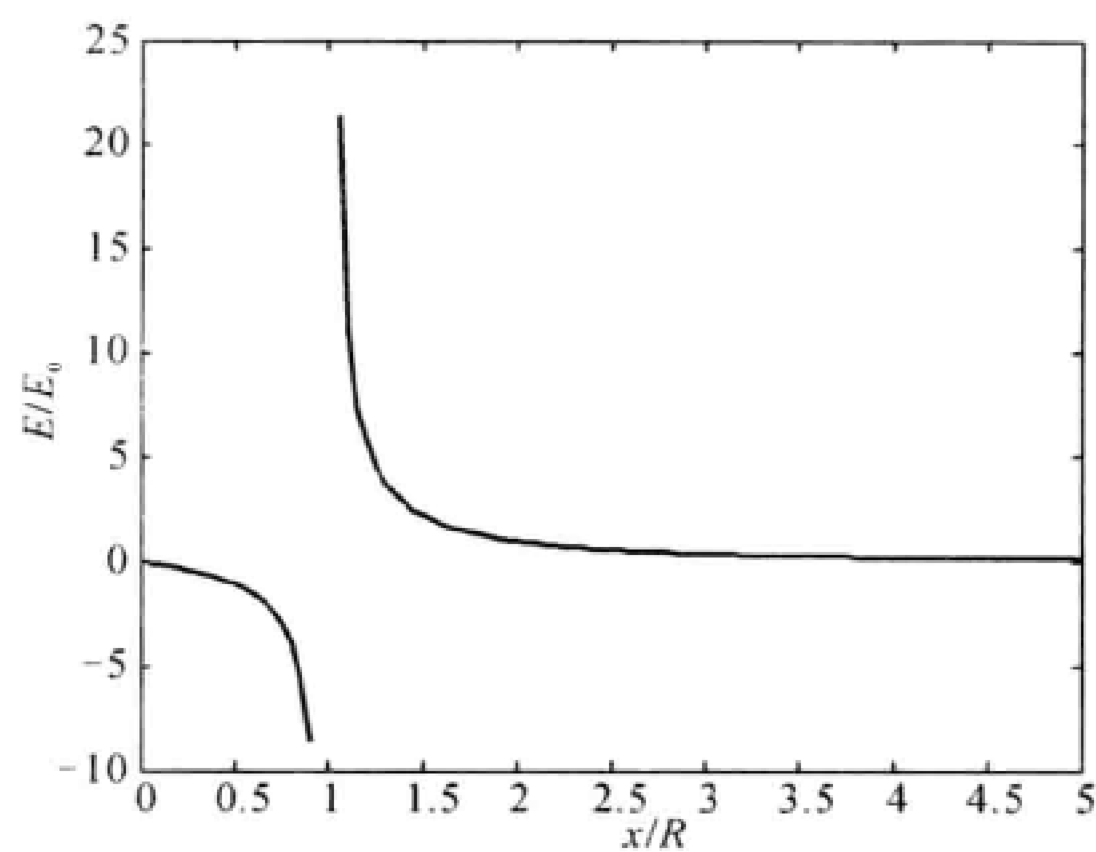


图 3.11

3.7.2 载流直线段的磁感应强度

根据毕奥-萨伐尔定律，真空中的通电导线 l 在点 P 产生的磁感应强度 $\boldsymbol{B}$ 为

$$\boldsymbol{B}=\frac{\mu_0}{4\pi}\int_l \frac{I\mathrm{d}\boldsymbol{l}\times\boldsymbol{r}_0}{r^2} \tag{3.88}$$

式中，μ_0 是真空的磁导率，$\boldsymbol{r}$ 是导线上的电流元 $I\mathrm{d}\boldsymbol{l}$ 到点 P 的位置矢量，$r=|\boldsymbol{r}|$，单位矢量 $\boldsymbol{r}_0=\boldsymbol{r}/r$，$I$ 是导线中的电流强度。

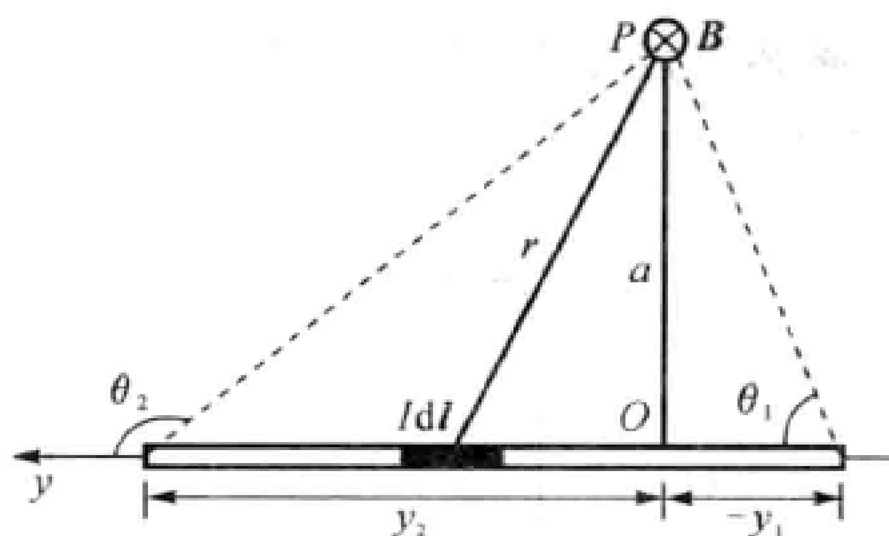


图　3.12

如图 3.12 所示的载流直导线在点 P 产生的磁感应强度 $\boldsymbol{B}$ 的大小为

$$B=\frac{\mu_0 I}{4\pi}\int_{y_1}^{y_2}\frac{a\mathrm{d}l}{(a^2+l^2)^{3/2}}=\frac{\mu_0 I}{4\pi a}\left[\frac{y_2}{(a^2+y_2^2)^{1/2}}-\frac{y_1}{(a^2+y_1^2)^{1/2}}\right] \tag{3.89}$$

式中，y_1 与 y_2 分别为直导线两端的坐标，a 为 P 点到直导线的距离。导线长度 $L=y_2-y_1$，借助于 $\alpha=-y_1/L$ 和 $\beta=a/L$，式(3.89) 重新写成

$$B=B_0\left(\frac{\beta}{2}\right)\int_{-\alpha}^{1-\alpha}\frac{\beta dz}{(\beta^2+z^2)^{3/2}}=B_0\left(\frac{1}{2}\right)\left[\frac{1-\alpha}{(\beta^2+(1-\alpha)^2)^{1/2}}+\frac{\alpha}{(\beta^2+\alpha^2)^{1/2}}\right] \tag{3.90}$$

其中，$B_0=\mu_0 I/(4\pi a)$。利用图中给出的两个角 θ_1 与 θ_2，式(3.89) 又能表示为[①]

$$B=\frac{\mu_0 I}{4\pi a}(\cos\theta_1-\cos\theta_2) \tag{3.91}$$

程序(3.12)　绘制平行于直导线的直线上磁感应强度分布曲线的 MATLAB 程序。

程序任务：取 $\beta=a/L$ 为固定值，利用变步长梯形积分方法计算 $\alpha=-y_1/L$ 取不同值时式(3.90) 中第一个等号后的积分表达式给出的 B，并与式(3.90) 中第二个等号后的结果进行比较。

```
global bt;
bt = 3;                          %β = a/L
af = -1/4:1/4:5/4                %α = -y1/L 的变化范围
n = length(af);
for I = 1:1:n
    BI(I) = (bt/2) * changtrap ('funMag', - af(I),1 - af(I),1e - 4);
          % 利用变步长梯形积分程序(3.5) 计算公式(3.90) 中第一个等号后公式表示的 B
end
af2 = -1/4:1/10:5/4;             %α = -y1/L
BF = (1/2). * ((1 - af2). /sqrt(bt.^2 + (1 - af2).^2) + af2. /sqrt(bt.^2 + af2.^2));
                                 % 计算公式(3.90) 中第二个等号后公式表示的 B
figure
plot(af - 0.5,BI,' * k',af2 - 0.5,BF, ' - k');     % 绘制磁场分布曲线
axis([- 0.8 0.8 0 1.1]);
xlabel('y/L');ylabel('B/B_o');
gtext('\beta = 3.00');                              % 注意 beta = bt
```

① 吴百诗. 大学物理基础(下册). 北京:科学出版社,2007.

对于不同的β值,运行程序的结果如图3.13所示。可见,当$a/L<0.10$时,在长度等于直导线长度一半的范围内,磁感应强度变化不大;当$a/L=1.0$时,在长度等于直导线长度的范围内,磁感应强度变化明显;当$a/L=3.0$时,在长度等于两倍直导线长度的范围内,磁感应强度基本不变。

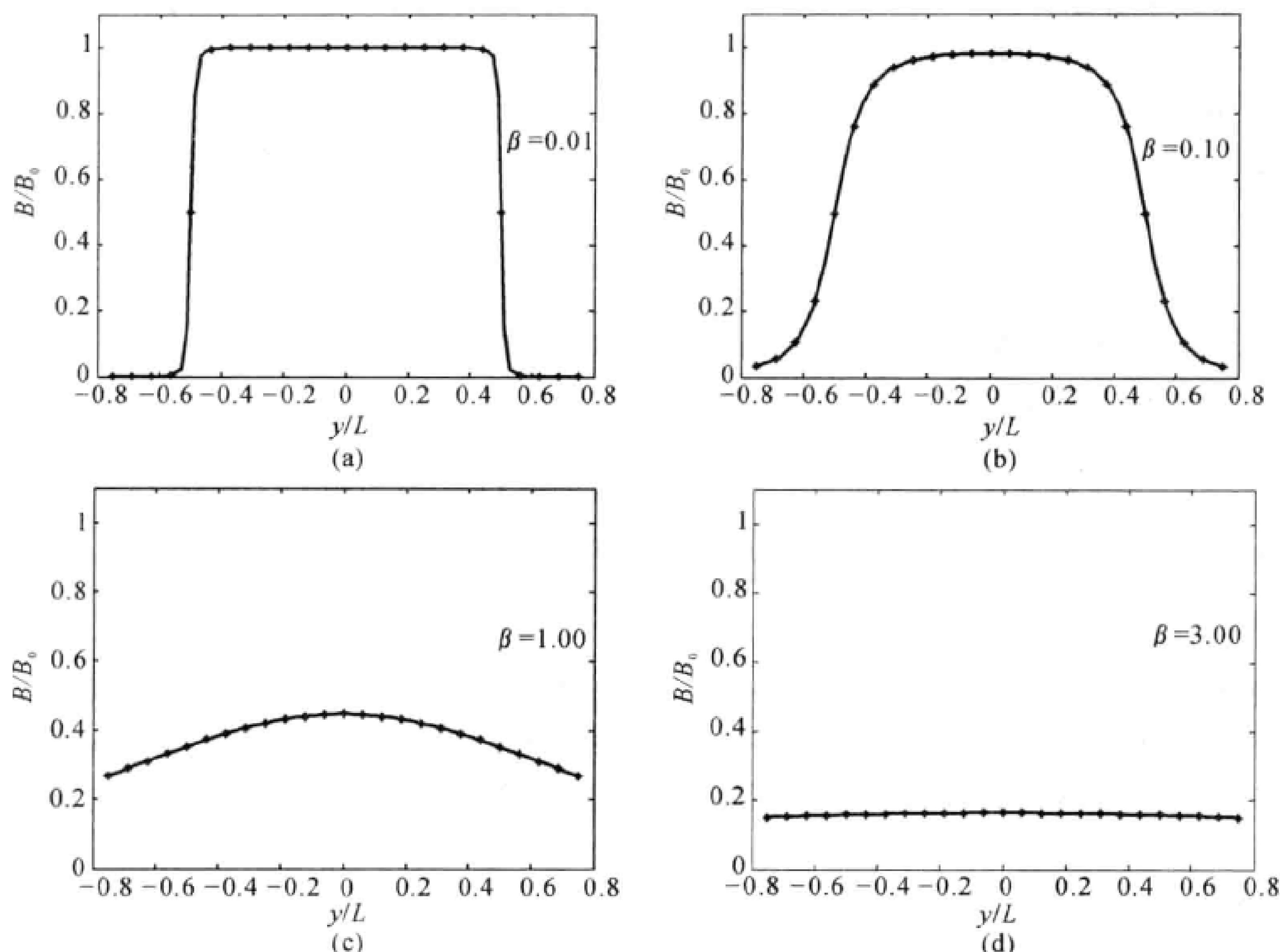


图 3.13

习 题

1. 确定下列求积公式中的常数,使其代数精度最高,并指出实际代数精度。

(1) $\int_0^2 f(x)\,\mathrm{d}x \approx A_0 f(0) + A_1 f(1) + A_2 f(2)$;

(2) $\int_{-1}^1 f(x)\,\mathrm{d}x \approx A[f(-1) + 2f(x_1) + 3f(x_2)]$。

2. 用代数精度的定义,直接验证辛普森公式具有3次代数精度。

3. 推导下列矩形积分公式的截断误差,并说明几何意义。

(1) $\int_a^b f(x)\,\mathrm{d}x \approx (b-a)f(a)$,截断误差 $R=\dfrac{f'(\xi)}{2}(b-a)^2, \xi\in(a,b)$;

(2) $\int_a^b f(x)\,\mathrm{d}x \approx (b-a)f(b)$,截断误差 $R=-\dfrac{f'(\xi)}{2}(b-a)^2, \xi\in(a,b)$;

(3) $\int_a^b f(x)\,\mathrm{d}x \approx (b-a)f\left(\dfrac{a+b}{2}\right)$,截断误差 $R=\dfrac{f''(\xi)}{24}(b-a)^3, \xi\in(a,b)$。

4. 若 $x \in [a,b]$，$f''(x) > 0$，证明用梯形求积公式计算积分 $\int_a^b f(x)\,\mathrm{d}x$ 所得结果比准确值大，并说明其几何意义。

5. 试分别用梯形公式、辛普森公式和柯特斯公式按五位小数计算积分 $\sqrt{\frac{2}{\pi}}\int_0^1 \mathrm{e}^{-\frac{x^2}{2}}\mathrm{d}x$。(准确值为 0.682 69)

6. 分别用复化梯形公式、复化辛普森公式和复化柯特斯公式计算下列积分：

(1) $\sqrt{\frac{2}{\pi}}\int_0^1 \mathrm{e}^{-\frac{x^2}{2}}\mathrm{d}x$；　　(要求将积分区间 4 等分)

(2) $\int_0^1 \frac{x}{4+x^2}\mathrm{d}x$；　　(要求将积分区间 8 等分)

(3) $\int_0^{2\pi} x\sin x\,\mathrm{d}x$；　　(要求将积分区间 8 等分)

7. 已知函数数据表，见表 3.8。

表　3.8

x_i	1.1	1.3	1.5
e^{x_i}	3.004 2	3.669 3	4.481 7

试分别用复化梯形公式和辛普森公式计算积分 $\int_{1.1}^{1.5}\mathrm{e}^x\,\mathrm{d}x$。

8. 已知 $\int_0^{0.8} f(x)\,\mathrm{d}x = 2$，且给出以下数据表(见表 3.9)：

表　3.9

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.8
$f(x)$	5	8	6	3	0	−3	−3	5

试用复化辛普森公式求 $f(0.7)$ 的近似值。

9. 若分别用复化梯形公式、复化辛普森公式和复化柯特斯公式计算 $\int_0^1 \mathrm{e}^{-x}\mathrm{d}x$，要求计算结果有 6 位有效数字，问步长应各取多少？

10. 用龙贝格方法计算下列积分：

(1) $\int_0^1 \sqrt{x}\,\mathrm{d}x$；　　(要求结果精确到 10^{-2})

(2) $\int_0^1 \frac{1}{1+x}\mathrm{d}x$；　　(要求二等分 4 次)

(3) $\int_0^{\pi} \mathrm{e}^x\cos x\,\mathrm{d}x$；　　(要求结果有 4 位有效数字)

11. 已知函数表，见表 3.10。

表 3.10

x	1.0	1.1	1.2	1.3	1.4
$f(x)$	0.250 0	0.226 8	0.206 6	0.189 0	0.173 6

用三点公式计算 $f'(1.0)$，$f'(1.1)$ 和 $f'(1.2)$。

12. 设 $f(x)=\exp(x)$ 。

(1) 步长 h 分别取 0.1，0.01，0.001，利用式(3.40) 计算 $f'(2.3)$ 的近似值，精度为小数点后 8 位或 9 位；

(2) 与 $f'(2.3)=\exp(2.3)$ 进行比较；

(3) 确定截断误差的边界，对三种情况均使用 $|f^{(3)}(c)|\leqslant\exp(2.4)\approx 11.023\ 176\ 38$。

13. 设 $f(x)=\sin x$，x 单位为 rad。

(1) 步长 h 分别取 0.1 和 0.01，利用式(3.43) 计算 $f'(0.8)$ 的近似值，并与 $f'(0.8)=\cos 0.8$ 进行比较；

(2) 利用式(3.44) 的外推公式计算(1) 中 $f'(0.8)$ 的近似值；

(3) 确定(1) 中截断误差的边界，对所有情况均使用 $|f^{(5)}(c)|\leqslant\cos 0.6\approx 0.825\ 335\ 615$。

14. 误差函数的定义为

$$\mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x \mathrm{e}^{-t^2}\,\mathrm{d}t$$

对区间进行 8 等分，分别用三种复化积分方法计算 $x=0.2$，0.3 时的函数值，并将结果与 MATLAB 内部函数 erf(x) 给出的结果进行比较。

15. 理想气体麦克斯韦速率分布函数为

$$f(v)=4\pi v^2\left(\frac{m}{2\pi kT}\right)^{3/2}\exp\left(-\frac{mv^2}{2kT}\right)$$

式中，m 为气体分子质量，k 为玻耳兹曼常量，T 为气体的开尔文温度。以最概然速率 $v_p=\sqrt{\frac{2kT}{m}}$ 能够将麦克斯韦速率分布函数表示为

$$f(v)=\frac{4}{\sqrt{\pi}}\left(\frac{v^2}{v_p^3}\right)\exp\left(-\frac{v^2}{v_p^2}\right)$$

已知室温(27℃) 下氢气分子的最概然速率 $v_p=1\ 578\mathrm{m/s}$，试用变步长梯形求积方法计算室温下氢气分子中：

(1) 速率在 $0\sim v_p$ 间隔内的分子数占总分子数的百分比；

(2) 速率在 $0\sim 3.3v_p$ 间隔内的分子数占总分子数的百分比；

(3) 速率在 $3\times10^4\sim 3\times10^8$ m/s 间隔内的分子数占总分子数的百分比。

要求精度为10^{-5}。

16. 菲涅耳积分在分析光的菲涅耳衍射时非常重要，它定义为

$$C(x)=\int_0^x \cos\frac{\pi t^2}{2}\mathrm{d}t,\quad S(x)=\int_0^x \sin\frac{\pi t^2}{2}\mathrm{d}t$$

试用龙贝格方法计算积分近似值，完成下列菲涅耳积分表(见表 3.11，要求精度为10^{-5})。

表　3.11

x	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5
$C(x)$									
$S(x)$									

17. 以大角度振动单摆的振动周期是

$$T=\frac{2T_0}{\pi}\int_0^{\frac{\pi}{2}}\frac{d\theta}{\sqrt{1-\sin^2(\theta_0/2)\sin^2\theta}}$$

式中，$T_0=2\pi\sqrt{l/g}$ 是摆角幅值 θ_0 很小时单摆振动的周期，m 为摆球质量，l 为摆长，g 为重力加速度。试分别用变步长梯形积分方法和龙贝格积分方法计算数值积分，完成关于单摆振动周期 T/T_0 与摆角幅值 θ_0 关系的表格（见表 3.12，要求精度为10^{-5}）。

表　3.12

$\theta_0/(°)$	1.0	3.0	5.0	7.0	10.0	15.0	20.0	25.0	30.0
T/T_0									

并确定 $|T-T_0|/T_0=0.05$ 时，θ_0 的近似值。

18. 电压 $E=E(t)$ 满足关系式 $E(t)=L(dI/dt)+RI(t)$，其中 R 是电阻，L 是电感。设 $R=2$，$L=0.05$，而且 $I(t)$ 的值如表 3.13 所示。

表　3.13

t/s	1.0	1.1	1.2	1.3	1.4
$I(t)$/A	8.227 7	7.242 8	5.990 8	4.526 0	2.912 2

(1) 通过数值微分求 $I'(1.2)$，并用它计算 $E(1.2)$；

(2) 比较计算结果与 $I(t)=10e^{-t/10}\sin(2t)$ 给出的准确值。

19. 一个物体的运动距离 $D=D(t)$ 如表 3.14 所示。

表　3.14

t/s	8.0	9.0	10.0	11.0	12.0
$D(t)$/m	17.453	21.460	25.752	30.301	35.084

(1) 通过数值微分求速率 $v(10)$；

(2) 比较计算结果与 $D(t)=-70+7t+70e^{-t/10}$ 给出的准确值。

20. 为了计算载流圆线圈（半径为 R，电流强度为 I）产生的磁场，建立如图 3.14 所示的柱坐标系。根据问题的轴对称性，选取计算的场点 P 在 xOy 坐标面上。点 $P(\rho,0,z)$ 处磁感应强度的三个分量分别为[①]

① 朱平. 圆电流空间磁场分布. 大学物理，2005，24(9)：13－17.

$$\left.\begin{aligned}
B_\rho &= \frac{\mu_0 I}{4\pi}\int_0^{2\pi}\frac{zR\cos\gamma\,\mathrm{d}\gamma}{(z^2+\rho^2+R^2-2\rho R\cos\gamma)^{3/2}} = \frac{\mu_0 I}{2\pi}\int_0^{\pi}\frac{zR\cos\gamma\,\mathrm{d}\gamma}{(z^2+\rho^2+R^2-2\rho R\cos\gamma)^{3/2}}\\
B_\varphi &= \frac{\mu_0 I}{4\pi}\int_0^{2\pi}\frac{zR\sin\gamma\,\mathrm{d}\gamma}{(z^2+\rho^2+R^2-2\rho R\cos\gamma)^{3/2}} = 0\\
B_z &= \frac{\mu_0 I}{4\pi}\int_0^{2\pi}\frac{R(R-\rho\cos\gamma)\,\mathrm{d}\gamma}{(z^2+\rho^2+R^2-2\rho R\cos\gamma)^{3/2}} = \frac{\mu_0 I}{2\pi}\int_0^{\pi}\frac{R(R-\rho\cos\gamma)\,\mathrm{d}\gamma}{(z^2+\rho^2+R^2-2\rho R\cos\gamma)^{3/2}}
\end{aligned}\right\} \tag{3.92}$$

经过推导，不为零的两个分量又表示为

$$\left.\begin{aligned}
B_\rho &= \frac{\mu_0 Izk'}{8\pi\rho\sqrt{\rho R}}\left(-2K(k')+\frac{2-k'^2}{1-k'^2}E(k')\right)\\
B_z &= \frac{\mu_0 Ik'}{8\pi\sqrt{\rho R}}\left(2K(k')+\frac{Rk'^2-(2-k'^2)\rho}{\rho(1-k'^2)}E(k')\right)
\end{aligned}\right\} \tag{3.93}$$

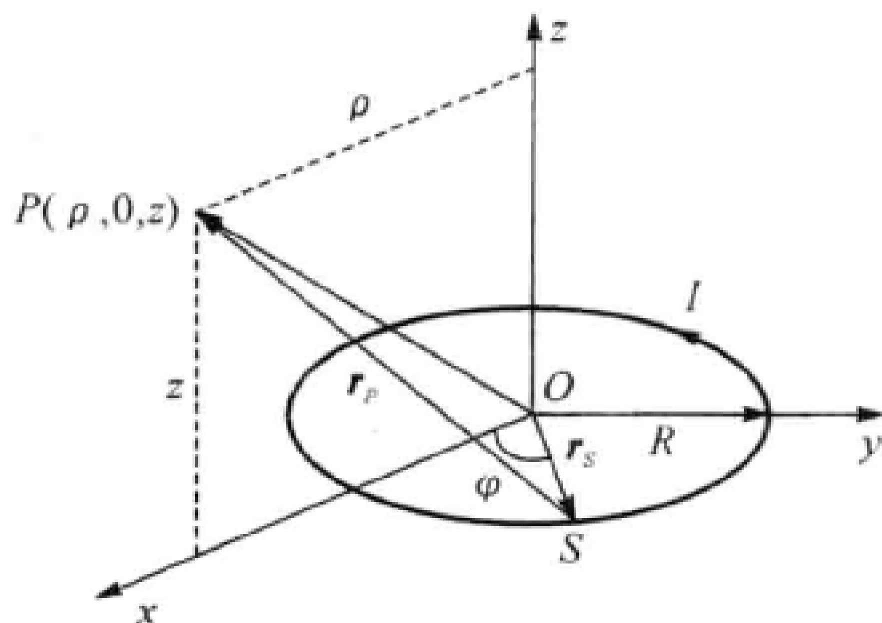


图 3.14

式中

$$k' = \left[\frac{4\rho R}{z^2+(\rho+R)^2}\right]^{1/2} = \left[\frac{4\rho/R}{(z/R)^2+(1+\rho/R)^2}\right]^{1/2} \tag{3.94}$$

显然，$k'<1$。函数 $K(k')$ 和 $E(k')$ 分别是第一类全椭圆积分和第二类全椭圆积分，由式(3.80) 和式(3.81) 给出。

若场点 P 位于圆线圈的对称轴（即 z 轴）上，则 $\rho=0, k'=0, K(0)=E(0)=\pi/2$，式(3.93) 简化为

$$\left.\begin{aligned}
B_\rho &= 0\\
B_z &= \frac{\mu_0 IR^2}{2(R^2+z^2)^{3/2}}
\end{aligned}\right\} \tag{3.95}$$

若场点 P 位于圆线圈平面内，即 $z=0$ 及

$$k' = \frac{2\sqrt{\rho/R}}{1+\rho/R} \tag{3.96}$$

式(3.93) 简化为

$$\left.\begin{aligned}
B_\rho &= 0\\
B_z &= \frac{B_z(O)k'}{4\pi\sqrt{\rho/R}}\left(2K(k')+\frac{k'^2-(2-k'^2)(\rho/R)}{(1-k'^2)(\rho/R)}E(k')\right)
\end{aligned}\right\} \tag{3.97}$$

其中，$B_z(O)=\mu_0 I/(2R)$，是圆线圈中心点 O 的磁感应强度。

根据以上公式，完成以下工作：

(1) 应用变步长积分方法计算积分，利用式(3.92) 分析圆线圈轴线上的磁感应强度 B_z 随场点坐标 z 的变化，并与解析表达式(3.95) 的结果进行比较；

(2) 应用龙贝格积分方法计算积分，利用式(3.92) 分析圆线圈平面内的磁感应强度 B_z 随场点坐标 ρ 的变化，并与式(3.97) 的计算结果进行比较。

21. 对于图 2.14 所示的单色光衍射系统，菲涅耳衍射公式为

$$U(x,y) = \frac{\exp(\mathrm{j}kz)}{\mathrm{j}\lambda z}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}U'(\xi,\eta)\exp\left\{\frac{\mathrm{j}k}{2z}\left[(x-\xi)^2+(y-\eta)^2\right]\right\}\mathrm{d}\xi\mathrm{d}\eta \tag{3.98}$$

夫琅禾费衍射公式为

$$U(x,y)=\frac{\exp(\mathrm{j}kz)\exp\left[\frac{\mathrm{j}k}{2z}(x^2+y^2)\right]}{\mathrm{j}\lambda z}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}U'(\xi,\eta)\exp\left[-\mathrm{j}\frac{2\pi}{\lambda z}(x\xi+y\eta)\right]\mathrm{d}\xi\mathrm{d}\eta \tag{3.99}$$

(1) 矩形孔的衍射。设衍射屏是带有矩形孔的不透光薄板，矩形孔的中心与坐标原点重合，四边分别与 ξ 轴与 η 轴平行，边长分别是 $2W_\xi$ 和 $2W_\eta$。在垂直于衍射屏的单位振幅平行光波照射下，透过衍射屏的光波复振幅为

$$U'(\xi,\eta)=\mathrm{rect}\left(\frac{\xi}{2W_\xi}\right)\mathrm{rect}\left(\frac{\eta}{2W_\eta}\right)=\begin{cases}1 & (|\xi|<W_\xi,|\eta|<W_\eta)\\ 0 & (\text{其他})\end{cases} \tag{3.100}$$

代入式(3.98)，得到矩形孔菲涅耳衍射的光场复振幅分布为

$$\begin{aligned}U(x,y)=\frac{\exp(\mathrm{j}kz)}{2\mathrm{j}}&\{[C(\alpha_2)-C(\alpha_1)]+\mathrm{j}[S(\alpha_2)-S(\alpha_1)]\}\cdot\\ &\{[C(\beta_2)-C(\beta_1)]+\mathrm{j}[S(\beta_2)-S(\beta_1)]\}\end{aligned} \tag{3.101}$$

式中①

$$C(x)=\int_0^x\cos\frac{\pi t^2}{2}\mathrm{d}t,\quad S(x)=\int_0^x\sin\frac{\pi t^2}{2}\mathrm{d}t$$

是菲涅耳积分函数，并且

$$\alpha_1=-\sqrt{\frac{2}{\lambda z}}(W_\xi+x),\quad \alpha_2=\sqrt{\frac{2}{\lambda z}}(W_\xi-x)$$

$$\beta_1=-\sqrt{\frac{2}{\lambda z}}(W_\eta+y),\quad \beta_2=\sqrt{\frac{2}{\lambda z}}(W_\eta-y)$$

把式(3.100)代入式(3.99)，得矩形孔夫琅禾费衍射的光场复振幅分布为

$$U(x,y)=\frac{4W_\xi W_\eta}{\mathrm{j}\lambda z}\exp(\mathrm{j}kz)\exp\left[\frac{\mathrm{j}k}{2z}(x^2+y^2)\right]\mathrm{sinc}\left(\frac{2W_\xi x}{\lambda z}\right)\mathrm{sinc}\left(\frac{2W_\eta y}{\lambda z}\right) \tag{3.102}$$

其中

$$\mathrm{sinc}(x)=\frac{\sin(\pi x)}{\pi x}$$

(2) 圆孔的衍射。当透过衍射屏的光场复振幅为 $U'(\xi,\eta)=U'(q)$ 时，即光场复振幅具有圆对称性，菲涅耳衍射积分式(3.98)改写为

$$U(r)=\frac{k}{\mathrm{j}z}\exp(\mathrm{j}kz)\exp\left(\frac{\mathrm{j}kr^2}{2z}\right)\int_0^\infty U'(q)\,\mathrm{J}_0\left(\frac{krq}{z}\right)\exp\left(\frac{\mathrm{j}kq^2}{2z}\right)q\,\mathrm{d}q \tag{3.103}$$

式中，$q=\sqrt{\xi^2+\eta^2}$，$r=\sqrt{x^2+y^2}$，$\mathrm{J}_0(x)$ 是零阶第一类贝塞尔函数。

设衍射屏是带有圆孔的不透光薄板，圆孔的中心与坐标原点重合，半径为 R。在垂直于衍射屏的单位振幅平面光波照射下，透过衍射屏的光波复振幅为

$$U'(q)=\mathrm{circ}\left(\frac{q}{R}\right)=\begin{cases}1 & (q<R)\\ 0 & (q>R)\end{cases} \tag{3.104}$$

代入式(3.103)，得到圆孔菲涅耳衍射的光场复振幅分布为

---

① 季家镕. 高等光学教程——光学的基本电磁理论. 北京：科学出版社，2007.

$$U(r)=\frac{k}{\mathrm{j}z}\exp(\mathrm{j}kz)\exp\left(\frac{\mathrm{j}kr^2}{2z}\right)\int_0^R \mathrm{J}_0\left(\frac{krq}{z}\right)\exp\left(\frac{\mathrm{j}kq^2}{2z}\right)q\,\mathrm{d}q \tag{3.105}$$

把式(3.104)代入式(3.99)，得到圆孔夫琅禾费衍射的光场复振幅分布为

$$U(r)=\frac{\pi R^2}{\mathrm{j}\lambda z}\exp(\mathrm{j}kz)\exp\left(\frac{\mathrm{j}kr^2}{2z}\right)\left[\frac{2\mathrm{J}_1\left(\frac{krR}{z}\right)}{\frac{krR}{z}}\right] \tag{3.106}$$

式中，函数 $\mathrm{J}_1(x)$ 是一阶第一类贝塞尔函数。

选择合适的数值积分方法，完成以下研究任务：

(1) 分别依据式(3.98)和式(3.101)，计算式(3.100)描述的矩形孔菲涅耳衍射光场的强度分布，并对依据式(3.98)和式(3.101)进行的数值计算过程进行比较；

(2) 依据式(3.99)，计算式(3.100)描述的矩形孔夫琅禾费衍射光场的强度分布，并与式(3.102)给出的结果进行比较；

(3) 依据式(3.105)，计算式(3.104)描述的圆孔菲涅耳衍射光场的强度分布，并将满足夫琅禾费近似条件情况下的计算结果与式(3.106)给出的结果进行比较；

(4) 依据式(3.99)，计算式(3.104)描述的圆孔夫琅禾费衍射光场的强度分布，并与式(3.106)给出的结果进行比较。

第 4 章 线性代数方程组的数值求解方法

在科学研究和工程计算中,许多问题的解决都涉及线性代数方程组的求解,例如电路分析、插值问题、函数拟合、非线性问题的线性化、有限差分法、有限元方法等等。因此,在数值计算方法中,线性代数方程组的求解具有基础性地位。

求解线性代数方程组的数值方法可以分为两大类,一类是直接法(也称精确解法),另一类是迭代法。所谓直接法,就是在不考虑舍入误差的情况下,通过有限步运算就可以得到方程组精确解的求解方法。本章介绍的高斯消去法和三角分解法是两类最基本的直接法,它们对求解以中低阶($n < 100$)稠密矩阵为系数矩阵的方程组非常有效。所谓迭代法,就是从选取的初始解向量出发,通过反复代入迭代公式,建立线性代数方程组的解向量序列,若解向量序列的极限存在,则这个极限就是线性方程组的精确解向量。迭代法较适用于求解系数矩阵为大型稀疏矩阵的方程组。由于在实际计算时,只能进行有限次迭代,所以即使不考虑舍入误差,也只能得到近似解。这里只介绍简单迭代法和赛德尔迭代法。

4.1 解线性方程组的直接法

关于未知量 $x_1, x_2, \cdots, x_n$ 的 n 元线性代数方程组通常表示为

$$\left.\begin{array}{c} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array}\right\} \tag{4.1}$$

其中,a_{ij} 和 $b_i (i, j = 1, 2, \cdots, n)$ 是常数。方程组式(4.1)可以简写为

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad (i = 1, 2, \cdots, n) \tag{4.2}$$

还可以表示为矩阵形式

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{4.3}$$

式中

$$\boldsymbol{A} = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{4.4}$$

称 $\boldsymbol{A}$ 为系数矩阵,$\boldsymbol{x}$ 为解向量,$\boldsymbol{b}$ 为常数向量。

当系数矩阵 $\boldsymbol{A}$ 的行列式 $D = \det(\boldsymbol{A}) \neq 0$(即 $\boldsymbol{A}$ 为非奇异方阵)时,由克莱姆(Cramer)法则,方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 的解唯一存在,它是

$$x_i = \frac{D_i}{D} \quad (i = 1, 2, \cdots, n) \tag{4.5}$$

式中，D_i 是用常数向量 $\boldsymbol{b}$ 代替系数矩阵 $\boldsymbol{A}$ 的第 i 列元素所得矩阵的行列式。

从理论上讲，用克莱姆法则总能求出线性方程组的精确解。但当方程组阶数较高时，求解计算量太大，计算机运行时间过长。用克莱姆法则解一个 n 阶方程组，需要计算 $n+1$ 个 n 阶行列式。一个 n 阶行列式有 $n!$ 项，每一项含有 n 个因子，因此计算一个 n 阶行列式需要进行 $n!(n-1)$ 次乘法，$n!-1$ 次加法。解一个 n 阶方程组，共进行乘法运算 $n!(n-1)(n+1)$ 次，加法运算 $(n!-1)(n+1)$ 次，除法运算 n 次。解一个 20 阶的线性方程组，单就乘法运算次数就达

$$20! \times (20-1)(20+1) \approx 9.707 \times 10^{20} \text{ 次}$$

用每秒运算 1 亿次的计算机计算，需要计算

$$\frac{9.707 \times 10^{20}}{10^8} = 9.707 \times 10^{12}\,\text{s} \approx 307\ 815 \text{ 年}$$

这是难以想象的。因此，必须研究线性代数方程组更为有效的数值求解方法。

4.1.1 三角方程组的求解方法

为了便于后面讨论，这里先介绍对角方程组、下三角方程组和上三角方程组等三角形方程组的解法。

1. 对角方程组

对角方程组的系数矩阵是对角矩阵，即

$$\boldsymbol{A} = \boldsymbol{D} = \begin{bmatrix} d_{11} & 0 & \cdots & 0 \\ 0 & d_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

方程式(4.2)简化为

$$d_{ii} x_i = b_i \quad (i = 1, 2, \cdots, n) \tag{4.6}$$

显然 $d_{ii} \neq 0$，方程组由 n 个独立的一元一次方程构成，其解为

$$x_i = \frac{b_i}{d_{ii}} \quad (i = 1, 2, \cdots, n) \tag{4.7}$$

2. 下三角方程组

下三角方程组的系数矩阵是下三角矩阵，即

$$\boldsymbol{A} = \boldsymbol{L} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \tag{4.8}$$

方程式(4.2)表示为

$$\sum_{j=1}^{i} l_{ij} x_j = b_i \quad (i = 1, 2, \cdots, n) \tag{4.9}$$

设 $l_{ii} \neq 0 (i = 1, 2, \cdots, n)$，从前 $i-1$ 个方程解出未知量 $x_1, x_2, \cdots, x_{i-1}$ 后，代入第 i 个方程

就能求出未知量 x_i。方程式(4.9)的解为

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} x_j}{l_{ii}} \quad (i = 1, 2, \cdots, n; \sum_{j=1}^{0} = 0) \tag{4.10}$$

3. 上三角方程组

上三角方程组的系数矩阵是上三角矩阵，即

$$\boldsymbol{A} = \boldsymbol{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \tag{4.11}$$

方程式(4.2)就是

$$\sum_{j=i}^{n} u_{ij} x_j = b_i \quad (i = 1, 2, \cdots, n) \tag{4.12}$$

设 $u_{ii} \neq 0 (i = 1, 2, \cdots, n)$，从第 $i+1$ 到第 n 个方程解出未知量 $x_{i+1}, x_{i+2}, \cdots, x_n$，代入第 i 个方程就能求出未知量 x_i。方程式(4.12)的解为

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} u_{ij} x_j}{u_{ii}} \quad (i = n, n-1, \cdots, 1; \sum_{j=n+1}^{n} = 0) \tag{4.13}$$

上面介绍的求解下三角方程组和上三角方程组的方法通常称为回代法。

程序(4.1)　用回代法解上三角方程组的 MATLAB 程序。

程序任务：用回代法解上三角方程组 $\boldsymbol{Ux} = \boldsymbol{b}$。

```
function x = uptriangle(U,b)
% 输入：U—— 上三角方程组的系数矩阵(N×N,非奇异)
%       b—— 上三角方程组的常数向量(N×1)
% 输出：x—— 上三角方程组的解向量(N×1)
n = length(b);          %n 是向量 b 的长度
x = zeros(n,1);         % 初始化解向量 x
x(n) = b(n)/U(n,n);     % 求出 x_n
for k = n-1:-1:1        % 回代求解
    x(k) = (b(k) - U(k,(k+1):n) * x((k+1):n))/U(k,k);  % 公式(4.13)
end
```

用程序(4.1)解方程组

$$\begin{cases} 3x_1 - 2x_2 + x_3 - x_4 = 8 \\ \quad 4x_2 - x_3 + 2x_4 = -3 \\ \quad\quad 2x_3 + 3x_4 = 11 \\ \quad\quad\quad 5x_4 = 15 \end{cases}$$

在 MATLAB 命令窗口中输入：

```
>> U=[3,-2,1,-1; 0,4,-1,2; 0,0,2,3; 0,0,0,5];    % 输入系数矩阵
>> b=[8,-3,11,15]';                               % 输入常数向量
```

```
>> x = uptriangle(U,b)                          % 调用程序(4.1)解方程组
```

输出结果：

```
x = [2 -2 1 3]'
```

即方程组的解向量是 $\boldsymbol{x} = [2\ -2\ 1\ 3]'$。

4.1.2 高斯消去法

高斯(Gauss)消去法是求解线性代数方程组的一种古老方法，目前它仍然是用计算机解以低阶稠密矩阵为系数矩阵方程组的有效方法。

高斯消去法分两步进行：① 消元过程。通过对方程组做初等变换(如方程组中两个方程交换位置、方程组中一个方程的若干倍与另一方程相加等)，逐步消去部分方程中的部分未知量，把方程组转化为同解的上三角方程组。② 回代过程。用回代法解上三角方程组。

1. 顺序高斯消去法

(1) 消元过程。为了符号统一，记 $a_{ij}^{(0)} = a_{ij}$，$a_{i,n+1}^{(0)} = b_i\ (i,j = 1,2,\cdots,n)$，则方程组式(4.1)改写为

$$\left.\begin{array}{l} a_{11}^{(0)} x_1 + a_{12}^{(0)} x_2 + \cdots + a_{1n}^{(0)} x_n = a_{1,n+1}^{(0)} \\ a_{21}^{(0)} x_1 + a_{22}^{(0)} x_2 + \cdots + a_{2n}^{(0)} x_n = a_{2,n+1}^{(0)} \\ \qquad\cdots\cdots \\ a_{n1}^{(0)} x_1 + a_{n2}^{(0)} x_2 + \cdots + a_{nn}^{(0)} x_n = a_{n,n+1}^{(0)} \end{array}\right\} \tag{4.14}$$

第 1 次消元：如果 $a_{11}^{(0)} \neq 0$，对第 1 个方程的各项均除以 $a_{11}^{(0)} = a_{11}$，则各项系数成为

$$a_{1j}^{(1)} = \frac{a_{1j}^{(0)}}{a_{11}^{(0)}} = \frac{a_{1j}}{a_{11}} \quad (j = 1,2,\cdots,n+1)$$

第 2 个到第 n 个方程分别减去 $a_{i1}^{(0)}\ (i = 2,3,\cdots,n)$ 乘变化后的第 1 个方程 $\sum_{j=1}^{n} a_{1j}^{(1)} x_j = a_{1,n+1}^{(1)}$，就消去了这些方程中的未知量 x_1，其余各项系数变成

$$a_{ij}^{(1)} = a_{ij}^{(0)} - a_{i1}^{(0)} a_{1j}^{(1)} = a_{ij}^{(0)} - a_{i1}^{(0)} \frac{a_{1j}^{(0)}}{a_{11}^{(0)}} \quad (i = 2,3,\cdots,n;\ j = 2,3,\cdots,n+1)$$

方程组式(4.14)转化为

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + \cdots + a_{1n}^{(1)} x_n = a_{1,n+1}^{(1)} \\ \qquad a_{22}^{(1)} x_2 + \cdots + a_{2n}^{(1)} x_n = a_{2,n+1}^{(1)} \\ \qquad\qquad\cdots\cdots \\ \qquad a_{n2}^{(1)} x_2 + \cdots + a_{nn}^{(1)} x_n = a_{n,n+1}^{(1)} \end{cases}$$

第 2 次消元：如果 $a_{22}^{(1)} \neq 0$，对第 2 个方程的各项均除以 $a_{22}^{(1)}$，各系数成为

$$a_{2j}^{(2)} = \frac{a_{2j}^{(1)}}{a_{22}^{(1)}} \quad (j = 2,3,\cdots,n+1)$$

第 3 个到第 n 个方程分别减去 $a_{i2}^{(1)}\ (i = 3,4,\cdots,n)$ 乘变化后的第 2 个方程 $\sum_{j=2}^{n} a_{2j}^{(2)} x_j = a_{2,n+1}^{(2)}$，就消去了这些方程中的未知量 x_2，其余各项系数表示为

$$a_{ij}^{(2)} = a_{ij}^{(1)} - a_{i2}^{(1)} a_{2j}^{(2)} = a_{ij}^{(1)} - a_{i2}^{(1)} \frac{a_{2j}^{(1)}}{a_{22}^{(1)}} \quad (i = 3,4,\cdots,n;\ j = 3,4,\cdots,n+1)$$

方程组又转化成

$$\begin{cases} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + \cdots + a_{1n}^{(1)} x_n = a_{1,n+1}^{(1)} \\ \qquad x_2 + a_{23}^{(2)} x_3 + \cdots + a_{2n}^{(2)} x_n = a_{2,n+1}^{(2)} \\ \qquad a_{33}^{(2)} x_3 + \cdots + a_{3n}^{(2)} x_n = a_{3,n+1}^{(2)} \\ \qquad \cdots\cdots \\ \qquad a_{n3}^{(2)} x_3 + \cdots + a_{nn}^{(2)} x_n = a_{n,n+1}^{(2)} \end{cases}$$

设第 $k-1$ 次消元后方程组是

$$\left.\begin{aligned} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + \cdots + a_{1n}^{(1)} x_n &= a_{1,n+1}^{(1)} \\ x_2 + a_{23}^{(2)} x_3 + \cdots + a_{2n}^{(2)} x_n &= a_{2,n+1}^{(2)} \\ \cdots\cdots \\ x_{k-1} + a_{k-1,k}^{(k-1)} x_k + \cdots + a_{k-1,n}^{(k-1)} x_n &= a_{k-1,n+1}^{(k-1)} \\ a_{kk}^{(k-1)} x_k + \cdots + a_{kn}^{(k-1)} x_n &= a_{k,n+1}^{(k-1)} \\ \cdots\cdots \\ a_{nk}^{(k-1)} x_k + \cdots + a_{nn}^{(k-1)} x_n &= a_{n,n+1}^{(k-1)} \end{aligned}\right\} \tag{4.15}$$

第 k 次消元($1 \leqslant k \leqslant n$):如果 $a_{kk}^{(k-1)} \neq 0$,对第 k 个方程的各项均除以 $a_{kk}^{(k-1)}$,各项系数变为

$$a_{kj}^{(k)} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}} \quad (j = k, k+1, \cdots, n+1) \tag{4.16}$$

第 $k+1$ 到第 n 个方程分别减去 $a_{ik}^{(k-1)}$ $(i = k+1, \cdots, n)$ 乘变化后的第 k 个方程 $\sum_{j=k}^{n} a_{kj}^{(k)} x_j = a_{k,n+1}^{(k)}$,就消去了方程中的未知量 x_k,其余各项系数表示为

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} a_{kj}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}} \quad (i = k+1, \cdots, n; j = k+1, \cdots, n+1) \tag{4.17}$$

方程组转化成

$$\left.\begin{aligned} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + \cdots + a_{1n}^{(1)} x_n &= a_{1,n+1}^{(1)} \\ x_2 + a_{23}^{(2)} x_3 + \cdots + a_{2n}^{(2)} x_n &= a_{2,n+1}^{(2)} \\ \cdots\cdots \\ x_k + a_{k,k+1}^{(k)} x_{k+1} + \cdots + a_{kn}^{(k)} x_n &= a_{k,n+1}^{(k)} \\ a_{k+1,k+1}^{(k)} x_{k+1} + \cdots + a_{k+1,n}^{(k)} x_n &= a_{k+1,n+1}^{(k)} \\ \cdots\cdots \\ a_{n,k+1}^{(k)} x_{k+1} + \cdots + a_{nn}^{(k)} x_n &= a_{n,n+1}^{(k)} \end{aligned}\right\} \tag{4.18}$$

经过 n 次消元后,方程组转化为上三角方程组

$$\left.\begin{aligned} x_1 + a_{12}^{(1)} x_2 + a_{13}^{(1)} x_3 + \cdots + a_{1n}^{(1)} x_n &= a_{1,n+1}^{(1)} \\ \cdots\cdots \\ x_k + a_{k,k+1}^{(k)} x_{k+1} + \cdots + a_{kn}^{(k)} x_n &= a_{k,n+1}^{(k)} \\ \cdots\cdots \\ x_n &= a_{n,n+1}^{(n)} \end{aligned}\right\} \tag{4.19}$$

把原方程组逐步化为上三角方程组的过程称为消元过程。在第 k 次($1 \leqslant k \leqslant n$) 消元过程中,元素 $a_{kk}^{(k-1)}$ 起着特殊的作用,故称它为主元素。

(2) 回代过程。按照解上三角方程组的公式(4.13),方程组式(4.19)的解为

$$x_n = a_{n,n+1}^{(n)}, \quad x_k = a_{k,n+1}^{(k)} - \sum_{j=k+1}^{n} a_{kj}^{(k)} x_j \quad (k = n-1, n-2, \cdots, 2, 1) \tag{4.20}$$

这种通过顺序消元和回代求解方程组的方法称为顺序高斯消去法,其特点是消去对角线下方的元素。

例 4.1 用顺序高斯消去法解方程组

$$\begin{cases} 2x_1 + x_2 + 4x_3 = -1 \\ 3x_1 + 2x_2 + x_3 = 4 \\ x_1 + 2x_2 + 4x_3 = -1 \end{cases}$$

解 第一次消元后,方程组是

$$\begin{cases} x_1 + 0.5x_2 + 2x_3 = -0.5 \\ 0.5x_2 - 5x_3 = 5.5 \\ 1.5x_2 + 2x_3 = -0.5 \end{cases}$$

第二次消元后,方程组成为

$$\begin{cases} x_1 + 0.5x_2 + 2x_3 = -0.5 \\ x_2 - 10x_3 = 11 \\ 17x_3 = -17 \end{cases}$$

第三次消元,对第三个方程除以 17,最终得到上三角方程组

$$\begin{cases} x_1 + 0.5x_2 + 2x_3 = -0.5 \\ x_2 - 10x_3 = 11 \\ x_3 = -1 \end{cases}$$

回带后得方程组的解是

$$x_3 = -1, \quad x_2 = 1, \quad x_1 = 1$$

主元素 $a_{kk}^{(k-1)} \neq 0 (k = 1, 2, \cdots, n)$ 是顺序高斯消去法能够顺利进行的保证,下面的定理给出了满足这一条件的基本要求。

定理(4.1) 用顺序高斯消去法解方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 时,主元素 $a_{kk}^{(k-1)} \neq 0 (k = 1, 2, \cdots, n)$ 的充分必要条件是系数矩阵 $\boldsymbol{A}$ 的所有顺序主子式均不等于零,即 $\Delta_i = \begin{vmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & & \vdots \\ a_{i1} & \cdots & a_{ii} \end{vmatrix} \neq 0 \ (i = 1, 2, \cdots, n)$。

从公式(4.16)、公式(4.17)和公式(4.20)可知,在顺序高斯消去法的消元过程中,乘法的运算次数是

$$\sum_{k=1}^{n-1} (n-k)^2 + \sum_{k=1}^{n-1} (n-k) = \frac{1}{3}(n^3 - n)$$

除法的运算次数是

$$\sum_{k=1}^{n-1} (n-k) = \frac{1}{2} n(n-1)$$

减法的运算次数是

$$\sum_{k=1}^{n-1}(n-k)^2+\sum_{k=1}^{n-1}(n-k)=\frac{1}{3}(n^3-n)$$

回代过程中乘法的运算次数为

$$\sum_{k=1}^{n-1}(n-k)=\frac{1}{2}n(n-1)$$

减法的运算次数是

$$\sum_{k=1}^{n-1}(n-k)=\frac{1}{2}n(n-1)$$

除掉主元素的次数为 n。因此,用顺序高斯消去法求解一个 n 阶线性方程组所需的乘除法运算次数为

$$\frac{1}{3}(n^3-n)+\frac{1}{2}n(n-1)+\frac{1}{2}n(n-1)+n=\frac{n^3}{3}+n^2-\frac{n}{3}$$

加减法的运算次数为

$$\frac{1}{3}(n^3-n)+\frac{1}{2}n(n-1)=\frac{n^3}{3}+\frac{n^2}{2}-\frac{5n}{6}$$

总的运算次数是

$$\left(\frac{n^3}{3}+n^2-\frac{n}{3}\right)+\left(\frac{n^3}{3}+\frac{n^2}{2}-\frac{5n}{6}\right)=\frac{2n^3}{3}+\frac{3n^2}{2}-\frac{7n}{6}$$

当 $n=20$ 时,用顺序高斯消去法只需计算 3 060 次乘除运算,运算总次数是 5 910。与使用克莱姆法则求解相比,计算量小很多。

2. 列主元素高斯消去法

要使顺序高斯消去法能按方程排列顺序完成消元,主元素只需满足 $a_{kk}^{(k-1)}\neq 0(k=1,2,\cdots,n)$。但在实际计算中,即使某些主元素 $|a_{kk}^{(k-1)}|$ 很小,用它作除数也会放大误差,为了避免这种不利现象的出现,人们提出了主元素消去法,这里只介绍列主元素高斯消去法。

列主元素高斯消去法的消元过程是:在顺序高斯消去法消元的每一步,选取与要消去的未知量在同一列上且绝对值最大的系数作为主元素进行消元。即按列选主元素,然后调换方程次序,使主元素处于对角线位置,再进行消元。

列主元素消去法的计算步骤:

(1) 消元过程。对第 $k-1$ 次消元所得方程组

$$\begin{cases}x_1+a_{12}^{(1)}x_2+a_{13}^{(1)}x_3+\cdots+a_{1n}^{(1)}x_n=a_{1,n+1}^{(1)}\\ \qquad x_2+a_{23}^{(2)}x_3+\cdots+a_{2n}^{(2)}x_n=a_{2,n+1}^{(2)}\\ \qquad\qquad\cdots\cdots\\ \qquad x_{k-1}+a_{k-1,k}^{(k-1)}x_k+\cdots+a_{k-1,n}^{(k-1)}x_n=a_{k-1,n+1}^{(k-1)}\\ \qquad\qquad a_{kk}^{(k-1)}x_k+\cdots+a_{kn}^{(k-1)}x_n=a_{k,n+1}^{(k-1)}\\ \qquad\qquad\cdots\cdots\\ \qquad\qquad a_{nk}^{(k-1)}x_k+\cdots+a_{nn}^{(k-1)}x_n=a_{n,n+1}^{(k-1)}\end{cases}$$

首先选择第 k 列元素 $a_{ik}^{(k-1)}(k\leqslant i\leqslant n)$ 的主元素 $a_{rk}^{(k-1)}$,使

$$|a_{rk}^{(k-1)}|=\max_{k\leqslant i\leqslant n}\{|a_{ik}^{(k-1)}|\}\tag{4.21}$$

如果 $a_{rk}^{(k-1)}=0$，说明方程组系数矩阵奇异，则停止计算。否则，交换第 r 个方程与第 k 个方程位置，之后利用公式(4.16)和公式(4.17)进行第 k 次消元。

(2) 回代过程。按照式(4.20)进行回代，就得到方程组的解。

例 4.2 在五位十进制的限制下，分别用顺序高斯消去法和列主元素高斯消去法解方程组

$$\begin{cases}0.000\,3x_1+3x_2=2.000\,1\\ x_1+x_2=1\end{cases}$$

解 (1) 用顺序高斯消去法解方程组。消元后方程组转化为上三角方程组

$$\begin{cases}x_1+10\,000x_2=6\,667\\ x_2=0.666\,67\end{cases}$$

回带后得出方程组的解是

$$x_2=0.666\,67,\quad x_1=0.3$$

(2) 用列主元素高斯消去法解方程组。交换两个方程位置，方程组转化为

$$\begin{cases}x_1+x_2=1\\ 0.000\,3x_1+3x_2=2.000\,1\end{cases}$$

消元后得到上三角方程组

$$\begin{cases}x_1+x_2=1\\ x_2=0.666\,67\end{cases}$$

回带后得出方程组的解为

$$x_2=0.666\,67,\quad x_1=0.333\,33$$

方程组的精确解是 $x_1=1/3$ 和 $x_2=2/3$。显然，用列主元素高斯消去法得到的解比用顺序高斯消去法得到的解误差更小。

程序(4.2) 用列主元素高斯消去法解方程组的 MATLAB 程序。

程序任务：用列主元素高斯消去法解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$。

```
function x = coluGauss(A,b)
% 输入:A—— 方程组的系数矩阵(N×N,非奇异)
%      b—— 方程组的常数向量(N×1)
% 输出:x—— 方程组的解向量(N×1)
n = length(b);      %n 是向量 x 的长度
A = [A b];          % 构建增广矩阵
% 下面的循环体完成矩阵 A 的上三角化
for k = 1:n-1
    [Mp p] = max(abs(A(k:n,k)));
        % 选出列阵 A(k ~ n, k) 的主元素,Mp 为其值,p 为其在列阵 A(k ~ n,k) 中的行号
    p = p+k-1;                          %p 值改变为主元素在(N×N) 矩阵中的行号
    if p > k
        T = A(k,:); A(k,:) = A(p,:);    A(p,:) = T; % 矩阵 A 的第 p 行与第 k 行交换
    end
% 以下两式完成消元
A((k+1):n,(k+1):(n+1)) = A((k+1):     n,(k+1):(n+1)) - A((k+1):n,k)/A(k,k) …
        * A(k,(k+1): (n+1));      % 计算(k+1) ~ n 行、(k+1) ~ (n+1) 列元素的新值
```

```
A((k+1):n,k) = zeros(n-k,1);          % 令(k+1) ~ n行、k列元素等于零
A                                     % 输出矩阵A对角化的中间值
end
% 以下用回代法解上三角方程组
x = zeros(n,1);                   % 初始化解向量x
x(n) = A(n,n+1)/A(n,n);           % 求出x_n
for k = n-1:-1:1                  % 回代求解
    x(k) = (A(k,n+1) - A(k,(k+1):n) * x((k+1):n))/A(k,k);
end
```

用程序(4.2) 解方程组

$$\begin{cases} x_1+2x_2+x_3+4x_4=13 \\ 2x_1+4x_3+3x_4=28 \\ 4x_1+2x_2+2x_3+x_4=20 \\ -3x_1+x_2+3x_3+2x_4=6 \end{cases}$$

在 MATLAB 命令窗口中输入：

```
>> A=[1,2,1,4; 2,0,4,3; 4,2,2,1; -3,1,3,2];     % 输入系数矩阵
>> b=[13,28,20,6]';                              % 输入常数向量
>> x=coluGauss(A,b)                              % 调用程序(4.2) 解方程组
```

输出结果：

```
A =
    4.0000    2.0000    2.0000    1.0000    20.0000
         0   -1.0000    3.0000    2.5000    18.0000
         0    1.5000    0.5000    3.7500     8.0000
         0    2.5000    4.5000    2.7500    21.0000
A =
    4.0000    2.0000    2.0000    1.0000    20.0000
         0    2.5000    4.5000    2.7500    21.0000
         0         0   -2.2000    2.1000    -4.6000
         0         0    4.8000    3.6000    26.4000
A =
    4.0000    2.0000    2.0000    1.0000    20.0000
         0    2.5000    4.5000    2.7500    21.0000
         0         0    4.8000    3.6000    26.4000
         0         0         0    3.7500     7.5000
x =[3 -1 4 2]'
```

即方程组的解向量是 $\boldsymbol{x}=[3\ -1\ 4\ 2]'$。

4.1.3　三角分解法

把一个方阵表示为几个三角矩阵乘积形式的过程称为矩阵的三角分解。三角分解法解方

程组的基本过程是：① 把线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 分解成两个三角矩阵的乘积 $\boldsymbol{LU}$ 形式(其中，$\boldsymbol{L}$ 为下三角矩阵，$\boldsymbol{U}$ 为上三角矩阵，公式 $\boldsymbol{A}=\boldsymbol{LU}$ 叫做方阵 $\boldsymbol{A}$ 的三角状分解或 $\boldsymbol{LU}$ 分解)，于是方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 就转化为 $\boldsymbol{LUx}=\boldsymbol{b}$；② 令 $\boldsymbol{Ux}=\boldsymbol{y}$，用解下三角方程组的方法从方程组 $\boldsymbol{Ly}=\boldsymbol{b}$ 解出中间解向量 $\boldsymbol{y}$；③ 再用解上三角方程组的方法从方程组 $\boldsymbol{Ux}=\boldsymbol{y}$ 解出原方程组的解 $\boldsymbol{x}$。

按照矩阵 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 的不同，常把解线性方程组的三角分解法分别称杜里特尔(Doolittle)分解法和克洛特(Crout)分解法。当 $\boldsymbol{L}$ 是单位下三角矩阵(对角线元素为 1 的下三角矩阵)、$\boldsymbol{U}$ 为上三角矩阵时，称为杜里特尔分解法；当 $\boldsymbol{L}$ 是下三角矩阵、$\boldsymbol{U}$ 是单位上三角矩阵(对角线元素为 1 的上三角矩阵)时，称为克洛特分解法。杜里特尔分解法和克洛特分解法统称为直接三角分解法。

定理(4.2) 设 $\boldsymbol{A}$ 为 n 阶方阵，如果 $\boldsymbol{A}$ 的各阶顺序主子式均不等于零，则矩阵 $\boldsymbol{A}$ 存在唯一的分解形式 $\boldsymbol{A}=\boldsymbol{L}_0\boldsymbol{D}\boldsymbol{U}_0$，其中，$\boldsymbol{L}_0$ 是单位下三角矩阵，$\boldsymbol{U}_0$ 为单位上三角矩阵，$\boldsymbol{D}$ 为 n 阶非奇异对角矩阵。进一步，若 $\boldsymbol{A}$ 还是对称矩阵，则 $\boldsymbol{U}_0=\boldsymbol{L}_0^{\mathrm{T}}$，即 $\boldsymbol{A}=\boldsymbol{L}_0\boldsymbol{D}\boldsymbol{L}_0^{\mathrm{T}}$。

1. 杜里特尔分解法

把方程式(4.3)的系数矩阵 $\boldsymbol{A}$ 分解为下面的三角形式：

$$\boldsymbol{A}=\boldsymbol{L}_0\boldsymbol{U} \tag{4.22}$$

其中

$$\boldsymbol{L}_0=\begin{bmatrix}1 & 0 & \cdots & 0\\ l_{21} & 1 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ l_{n1} & l_{n2} & \cdots & 1\end{bmatrix},\quad \boldsymbol{U}=\begin{bmatrix}u_{11} & u_{12} & \cdots & u_{1n}\\ 0 & u_{22} & \cdots & u_{2n}\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & u_{nn}\end{bmatrix} \tag{4.23}$$

分别是单位下三角矩阵和上三角矩阵。通过矩阵乘法运算和元素对比可知，三角矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{L}_0$ 的元素分别表示为

$$u_{ij}=a_{ij}-\sum_{r=1}^{i-1}l_{ir}u_{rj}\quad\left(i=1,2,\cdots,n;j=i,i+1,\cdots,n;\sum_{r=1}^{0}=0\right) \tag{4.24}$$

$$l_{ij}=\frac{a_{ij}-\sum\limits_{r=1}^{j-1}l_{ir}u_{rj}}{u_{jj}}\quad\left(j=1,\cdots,n-1;i=j+1,\cdots,n;\sum_{r=1}^{0}=0\right) \tag{4.25}$$

分解系数矩阵的计算步骤是：① 用公式(4.24)计算 $\boldsymbol{U}$ 的第一行元素 $u_{1j}=a_{1j}(j=1,2,\cdots,n)$，用公式(4.25)计算 $\boldsymbol{L}_0$ 的第一列元素 $l_{i1}=a_{i1}/u_{11}(i=2,3,\cdots,n)$；② 用公式(4.24)计算 $\boldsymbol{U}$ 的第二行元素，用公式(4.25)计算 $\boldsymbol{L}_0$ 的第二列元素，持续进行，直到计算出所有元素。计算顺序归纳于图 4.1，其中带圈的数字表示计算顺序。

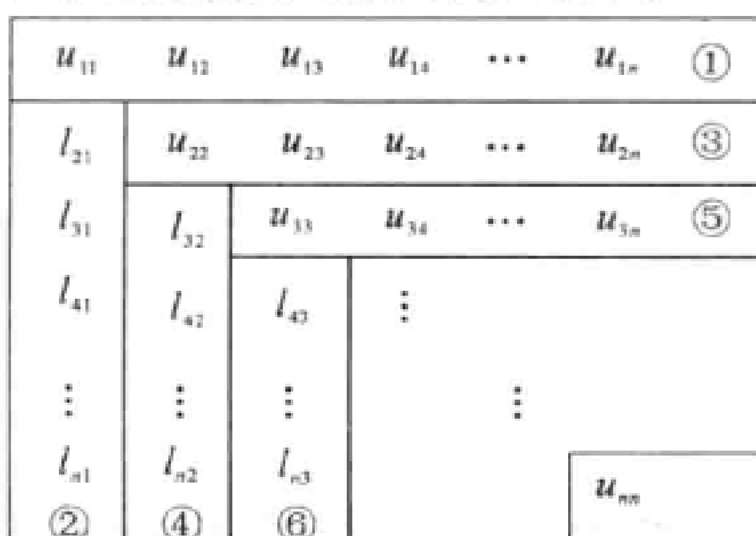


图 4.1

解方程组 $\boldsymbol{L}_0\boldsymbol{U}\boldsymbol{x}=\boldsymbol{b}$ 的步骤如下:

(1) 令 $\boldsymbol{U}\boldsymbol{x}=\boldsymbol{y}$,解方程组 $\boldsymbol{L}_0\boldsymbol{y}=\boldsymbol{b}$。根据公式(4.10),得

$$y_i=b_i-\sum_{r=1}^{i-1}l_{ir}y_r \quad (i=1,2,\cdots,n;\sum_{r=1}^{0}=0)$$

(2) 解方程组 $\boldsymbol{U}\boldsymbol{x}=\boldsymbol{y}$。根据公式(4.13),有

$$x_i=\frac{y_i-\sum\limits_{r=i+1}^{n}u_{ir}x_r}{u_{ii}} \quad (i=n,n-1,\cdots,1;\sum_{r=n+1}^{n}=0) \tag{4.26}$$

例 4.3　用杜里特尔分解法解方程组

$$\begin{cases}2x_1+4x_2+2x_3+6x_4=9\\4x_1+9x_2+6x_3+15x_4=23\\2x_1+6x_2+9x_3+18x_4=22\\6x_1+15x_2+18x_3+40x_4=47\end{cases}$$

解　方程组的系数矩阵和常数向量分别是

$$\boldsymbol{A}=\begin{bmatrix}2&4&2&6\\4&9&6&15\\2&6&9&18\\6&15&18&40\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}9\\23\\22\\47\end{bmatrix}$$

(1) 对系数矩阵进行 $\boldsymbol{A}=\boldsymbol{L}_0\boldsymbol{U}$ 分解,得到

$$\boldsymbol{L}_0=\begin{bmatrix}1&&&\\2&1&&\\1&2&1&\\3&3&2&1\end{bmatrix},\quad \boldsymbol{U}=\begin{bmatrix}2&4&2&6\\&1&2&3\\&&3&6\\&&&1\end{bmatrix}$$

(2) 解方程组$\boldsymbol{L}_0\boldsymbol{y}=\boldsymbol{b}$,即

$$\begin{bmatrix}1&&&\\2&1&&\\1&2&1&\\3&3&2&1\end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix}=\begin{bmatrix}9\\23\\22\\47\end{bmatrix}$$

有

$$\boldsymbol{y}=[9\quad 5\quad 3\quad -1]^{\mathrm{T}}$$

(3) 解方程组 $\boldsymbol{U}\boldsymbol{x}=\boldsymbol{y}$,即

$$\begin{bmatrix}2&4&2&6\\&1&2&3\\&&3&6\\&&&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}9\\5\\3\\-1\end{bmatrix}$$

得出原方程组的解为

$$\boldsymbol{x}=[0.5\quad 2\quad 3\quad -1]^{\mathrm{T}}$$

程序(4.3)　非奇异方阵杜里特尔分解的 MATLAB 程序。

程序任务:对非奇异方阵 $\boldsymbol{A}$ 进行杜里特尔分解,给出矩阵$\boldsymbol{L}_0$ 和 $\boldsymbol{U}$。

```
function [L U] = LUDoolittle(A)
% 输入:A—— 非奇异方阵(N×N)
% 输出:L—— 单位下三角矩阵(N×N)
%        U—— 上三角矩阵(N×N)
n = length(A);                              %n是方阵A的行数和列数
L = eye(n); U = zeros(n);                   % 初始化方阵L和U
U(1,:) = A(1,:); L(2:n,1) = A(2:n,1)/U(1,1); % 计算U的第一行元素和L的第一列元素
for k = 2:n
    U(k,k:n) = A(k,k:n) - L(k,1:(k-1)) * U(1:(k-1),k:n);
                                            % 计算U的第k行、k~n列元素
    L((k+1):n,k) = (A((k+1):n,k) - L((k+1):n,1:(k-1)) * U(1:(k-1),k))/U(k,k);
                                            % 计算L的第k列、(k+1)~n行元素
end
```

用程序(4.3)对方阵

$$
\boldsymbol{A}=\begin{bmatrix} 1 & 1 & 0 & 4 \\ 2 & -1 & 5 & 0 \\ 5 & 2 & 1 & 2 \\ -3 & 0 & 2 & 6 \end{bmatrix}
$$

进行杜里特尔分解。

在MATLAB命令窗口中输入:

```
>> A=[1,1,0,4; 2,-1,5,0; 5,2,1,2; -3,0,2,6];     % 输入方阵A
>> [L U]=LUDoolittle(A)                           % 调用程序(4.3)完成分解
```

输出结果:

```
L =
     1.0000         0         0         0
     2.0000    1.0000         0         0
     5.0000    1.0000    1.0000         0
    -3.0000   -1.0000   -1.7500    1.0000
U =
    1.0000    1.0000         0     4.0000
         0   -3.0000    5.0000    -8.0000
         0         0   -4.0000   -10.0000
         0         0         0    -7.5000
```

程序(4.4) 用杜里特尔分解法解方程组的MATLAB程序。

程序任务:用杜里特尔分解法解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$。

```
function x = EqsDoolittle(A, b)
% 输入:A—— 方程组的系数矩阵(N×N,非奇异)
%        b—— 方程组的常数向量(N×1)
% 输出:x—— 方程组的解向量(N×1)
```

```
n = length(A);                %n 是方阵 A 的行数和列数
% 以下程序段对矩阵 A 进行杜里特尔分解
L = eye(n); U = zeros(n);     % 初始化方阵 L 和 U
U(1,:) = A(1,:); L(2:n,1) = A(2:n,1)/U(1,1);      % 计算 U 的第一行元素和 L 的第一列元素
for k = 2:n
    U(k,k:n) = A(k,k:n) - L(k,1:(k-1)) * U(1:(k-1),k:n);
                                                  % 计算 U 的第 k 行、k ~ n 列元素
    L((k+1):n,k) = (A((k+1):n,k) - L((k+1):n,1:(k-1)) * U(1:(k-1),k))/U(k,k);
                                                  % 计算 L 的第 k 列、(k+1) ~ n 行元素
end
% 以下程序段解方程组 L。y = b
y = zeros(n,1);      % 初始化向量 y
y(1) = b(1);         % 计算 y(1)
for i = 2:n
    y(i) = b(i) - L(i,1:(i-1)) * y(1:(i-1));     % 计算 y(2 ~ n)
end
% 以下程序段解方程组 Ux = y
x = zeros(n,1);      % 初始化向量 x
x(n) = y(n)/U(n,n); % 计算 x(n)
for i = (n-1):-1:1
    x(i) = (y(i) - U(i, (i+1):n) * x((i+1):n))/U(i,i);     % 计算 x((n-1) ~ 1)
end
```

用程序(4.4)解方程组

$$
\begin{cases}
4x_1 + 8x_2 + 4x_3 = 8 \\
x_1 + 5x_2 + 4x_3 - 3x_4 = -4 \\
x_1 + 4x_2 + 7x_3 + 2x_4 = 10 \\
x_1 + 3x_2 - 2x_4 = -4
\end{cases}
$$

在 MATLAB 命令窗口输入：

```
>> A = [4,8,4,0;1,5,4,-3;1,4,7,2;1,3,0,-2];     % 输入系数矩阵
>> b = [8,-4,10,-4]';                            % 输入常数向量
>> x = EqsDoolittle(A,b)                         % 调用程序(4.4)解方程组
```

输出结果：

x = [3 −1 1 2]′

即方程组的解是 $\boldsymbol{x} = [3\ -1\ 1\ 2]'$。

2. *克洛特分解法*

把方程式(4.3)系数矩阵 $\boldsymbol{A}$ 分解成

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}_0 \tag{4.27}$$

其中

$$
\boldsymbol{L}=\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}, \quad \boldsymbol{U}_0=\begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ 0 & 1 & \cdots & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{4.28}
$$

分别是下三角矩阵和单位上三角矩阵，通过矩阵乘法运算和元素对比可知，它们的元素分别表示为

$$
l_{ij}=a_{ij}-\sum_{r=1}^{j-1} l_{ir} u_{rj} \quad (j=1,2,\cdots,n; i=j,j+1,\cdots,n; \sum_{r=1}^{0}=0) \tag{4.29}
$$

$$
u_{ij}=\frac{a_{ij}-\sum\limits_{r=1}^{i-1} l_{ir} u_{rj}}{l_{ii}} \quad (i=1,\cdots,n-1; j=i+1,\cdots,n; \sum_{r=1}^{0}=0) \tag{4.30}
$$

系数矩阵分解的计算步骤是：① 用公式(4.29) 计算 $\boldsymbol{L}$ 的第一列元素 $l_{i1}=a_{i1}(i=1,2,\cdots,n)$，用公式(4.30) 计算 $\boldsymbol{U}_0$ 的第一行元素 $u_{1j}=a_{1j}/l_{11}(j=2,3,\cdots,n)$；② 用公式(4.29) 计算 $\boldsymbol{L}$ 的第二列元素，再用公式(4.30) 计算 $\boldsymbol{U}_0$ 的第二行元素，继续进行，直到计算出所有元素。计算顺序归纳于图 4.2，其中带圈的数字表示计算顺序。

l_{11}	u_{12}	u_{13}	u_{14}	$\cdots$	u_{1n}	②
l_{21}	l_{22}	u_{23}	u_{24}	$\cdots$	u_{2n}	④
l_{31}	l_{32}	l_{33}	u_{34}	$\cdots$	u_{3n}	⑥
l_{41}	l_{42}	l_{43}	$\vdots$			
$\vdots$	$\vdots$	$\vdots$		$\vdots$		
l_{n1}	l_{n2}	l_{n3}			l_{nn}	
①	③	⑤				

图 4.2

解方程组 $\boldsymbol{L}\boldsymbol{U}_0\boldsymbol{x}=\boldsymbol{b}$ 的步骤如下：

(1) 令 $\boldsymbol{U}_0\boldsymbol{x}=\boldsymbol{y}$，解方程组 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b}$。根据公式(4.10)，得

$$
y_i=\frac{b_i-\sum\limits_{r=1}^{i-1} l_{ir} y_r}{l_{ii}} \quad (i=1,2,\cdots,n; \sum_{r=1}^{0}=0) \tag{4.31}
$$

(2) 解方程组 $\boldsymbol{U}_0\boldsymbol{x}=\boldsymbol{y}$，根据公式(4.13)，有

$$
x_i=y_i-\sum_{r=i+1}^{n} u_{ir} x_r \quad (i=n,n-1,\cdots,1; \sum_{r=n+1}^{n}=0) \tag{4.32}
$$

例 4.4 用克洛特分解法解方程组

$$
\begin{bmatrix} 2 & 4 & 2 & 6 \\ 4 & 5 & -5 & 9 \\ 3 & 8 & 5 & 3 \\ 1 & 5 & 8 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 15 \\ 3 \\ 3 \end{bmatrix}
$$

解 (1) 对系数矩阵进行 $\boldsymbol{A}=\boldsymbol{L}\,\boldsymbol{U}_0$ 分解，得到

$$\boldsymbol{L}=\begin{bmatrix}2&&&\\4&-3&&\\3&2&-4&\\1&3&-2&5\end{bmatrix},\quad \boldsymbol{U}_0=\begin{bmatrix}1&2&1&3\\&1&3&1\\&&1&2\\&&&1\end{bmatrix}$$

(2) 解方程组 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b}$，即

$$\begin{bmatrix}2&&&\\4&-3&&\\3&2&-4&\\1&3&-2&5\end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix}=\begin{bmatrix}6\\15\\3\\3\end{bmatrix}$$

有

$$\boldsymbol{y}=[3\quad -1\quad 1\quad 1]^{\mathrm{T}}$$

(3) 解方程组 $\boldsymbol{U}_0\boldsymbol{x}=\boldsymbol{y}$，即

$$\begin{bmatrix}1&2&1&3\\&1&3&1\\&&1&2\\&&&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}3\\-1\\1\\1\end{bmatrix}$$

得到原方程组的解为

$$\boldsymbol{x}=[-1\quad 1\quad -1\quad 1]^{\mathrm{T}}$$

程序(4.5)　非奇异方阵克洛特分解的 MATLAB 程序。

程序任务：对非奇异方阵 $\boldsymbol{A}$ 进行克洛特分解，给出矩阵 $\boldsymbol{L}$ 和 $\boldsymbol{U}_0$。

```
function [L U] = LUCrout(A)
% 输入:A—— 非奇异方阵(N×N)
% 输出:L—— 下三角矩阵(N×N)
%        U—— 单位上三角矩阵(N×N)
n = length(A);                    %n 是方阵 A 的行数和列数
L = zeros(n); U = eye(n);         % 初始化方阵 L 和 U
L(:,1) = A(:,1); U(1,2:n) = A(1,2:n)/L(1,1);        % 计算 L 的第一列元素和 U 的第一行元素
for k = 2:n
    L(k:n,k) = A(k:n,k) - L(k:n,1:(k-1)) * U(1:(k-1),k);
                                                    % 计算 U 的第 k 列、k ~ n 行元素
    U(k,(k+1):n) = (A(k,(k+1):n) - L(k,1:(k-1)) * U(1:(k-1), (k+1):n))/L(k,k);
                                                    % 计算 U 的第 k 行、(k+1) ~ n 行元素
end
```

用程序(4.5) 对方阵

$$\boldsymbol{A}=\begin{bmatrix}1&1&0&4\\2&-1&5&0\\5&2&1&2\\-3&0&2&6\end{bmatrix}$$

进行克洛特分解。

在 MATLAB 命令窗口中输入：

```
>> A=[1,1,0,4; 2,-1,5,0; 5,2,1,2; -3,0,2,6];     % 输入方阵 A
>> [L U]=LUCrout(A)                              % 调用程序(4.5)完成分解
```

输出结果：

```
L =
     1.0000         0         0         0
     2.0000   -3.0000         0         0
     5.0000   -3.0000   -4.0000         0
    -3.0000    3.0000    7.0000   -7.5000
U =
    1.0000    1.0000         0    4.0000
         0    1.0000   -1.6667    2.6667
         0         0    1.0000    2.5000
         0         0         0    1.0000
```

程序(4.6) 用克洛特分解法解方程组的 MATLAB 程序。

程序任务：用克洛特分解法解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$。

```
function x = EqsCrout (A,b)
% 输入:A—— 方程组的系数矩阵(N×N,非奇异)
%      b—— 方程组的常数向量(N×1)
% 输出:x—— 方程组的解向量(N×1)
% 以下程序段对矩阵 A 进行克洛特分解
n = length(A);                %n 是方阵 A 的行数和列数
L = zeros(n); U = eye(n);     % 初始化方阵 L 和 U
L(:,1) = A(:,1); U(1,2:n) = A(1,2:n)/L(1,1);
                              % 计算 L 的第一列元素和 U 的第一行元素
for k = 2:n
    L(k:n,k) = A(k:n,k) - L(k:n,1:(k-1)) * U(1:(k-1),k);
                              % 计算 U 的第 k 列、k ~ n 行元素
    U(k,(k+1):n) = (A(k,(k+1):n) - L(k,1:(k-1)) * U(1:(k-1), (k+1):n))/L(k,k);
                              % 计算 U 的第 k 行、(k+1) ~ n 行元素
end
% 以下程序段解方程组 Ly = b
y = zeros(n,1);          % 初始化向量 y
y(1) = b(1)/L(1,1);      % 计算 y(1)
for i = 2:n
    y(i) = (b(i) - L(i,1:(i-1)) * y(1:(i-1)))/L(i,i);% 计算 y(2 ~ n)
end
% 以下程序段解方程组 U。x = y
x = zeros(n,1);                              % 初始化向量 x
```

```
x(n) = y(n);                                   % 计算 x(n)
for i = (n-1):-1:1
    x(i) = y(i) - U(i,(i+1):n) * x((i+1):n);% 计算 x((n-1) ~ 1)
end
```

用程序(4.6)解方程组

$$\begin{cases}4x_1+8x_2+4x_3=8\\ x_1+5x_2+4x_3-3x_4=-4\\ x_1+4x_2+7x_3+2x_4=10\\ x_1+3x_2-2x_4=-4\end{cases}$$

在 MATLAB 命令窗口输入：

```
>> A=[4,8,4,0; 1,5,4,-3; 1,4,7,2; 1,3,0,-2];    % 输入系数矩阵
>> b=[8,-4,10,-4]';                                % 输入常数向量
>> x=EqsCrout(A,b)                                 % 调用程序(4.6)解方程组
```

输出结果：

```
x=[3 -1 1 2]'
```

即方程组的解是 $\boldsymbol{x}=[3\ -1\ 1\ 2]'$。

3. 平方根方法

定理(4.3)　如果 $\boldsymbol{A}$ 为 n 阶正定对称矩阵，则存在一个实的非奇异下三角矩阵 $\boldsymbol{L}$，使得 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$。当限定 $\boldsymbol{L}$ 的对角元素全为正时，这种分解是唯一的。

正定对称矩阵 $\boldsymbol{A}$ 的三角分解 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 称为乔列斯基(Cholesky)分解。

利用系数矩阵的乔列斯基分解，求解对称正定方程组的方法称为平方根方法。其基本过程如下：

(1) 对系数矩阵进行乔列斯基分解 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$，这里下三角矩阵为

$$\boldsymbol{L}=\begin{bmatrix}l_{11} & 0 & \cdots & 0\\ l_{21} & l_{22} & \cdots & 0\\ \vdots & \vdots & & \vdots\\ l_{n1} & l_{n2} & \cdots & l_{nn}\end{bmatrix} \tag{4.33}$$

其中

$$l_{ii}=\left(a_{ii}-\sum_{r=1}^{i-1}l_{ir}^2\right)^{1/2}\quad\left(i=1,2,\cdots,n;\ \sum_{r=1}^{0}=0\right)$$

$$l_{ij}=\frac{a_{ij}-\sum_{r=1}^{j-1}l_{ir}l_{jr}}{l_{jj}}\quad(j=1,2,\cdots,n;i=j+1,\cdots,n) \tag{4.34}$$

(2) 解方程组 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b}$，有

$$y_i=\frac{b_i-\sum_{r=1}^{i-1}l_{ir}y_r}{l_{ii}}\quad\left(i=1,2,\cdots,n;\ \sum_{r=1}^{0}=0\right) \tag{4.35}$$

(3) 解方程组$\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{y}$,得

$$x_i=\frac{y_i-\sum\limits_{r=i+1}^{n} l_{ri}x_r}{l_{ii}}\quad (i=n,n-1,\cdots,1;\ \sum_{r=n+1}^{n}=0)\tag{4.36}$$

例 4.5 用平方根方法解方程组

$$\begin{bmatrix}4 & 2 & -2\\ 2 & 2 & -3\\ -2 & -3 & 14\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}10\\ 5\\ 4\end{bmatrix}$$

解 由于

$$a_{11}=4>0,\quad \begin{vmatrix}4 & 2\\ 2 & 2\end{vmatrix}=4>0,\quad \begin{vmatrix}4 & 2 & -2\\ 2 & 2 & -3\\ -2 & -3 & 14\end{vmatrix}=36>0$$

因此方程组的系数矩阵是对称正定的。对其进行$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$,得出下三角矩阵

$$\boldsymbol{L}=\begin{bmatrix}2 & & \\ 1 & 1 & \\ -1 & -2 & 3\end{bmatrix}$$

解方程组 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{b}$,即

$$\begin{bmatrix}2 & & \\ 1 & 1 & \\ -1 & -2 & 3\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ y_3\end{bmatrix}=\begin{bmatrix}10\\ 5\\ 4\end{bmatrix}$$

有

$$\boldsymbol{y}=[5\quad 0\quad 3]^{\mathrm{T}}$$

解方程组$\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{y}$,即

$$\begin{bmatrix}2 & 1 & -1\\ & 1 & -2\\ & & 3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}5\\ 0\\ 3\end{bmatrix}$$

最终求出原方程组的解

$$\boldsymbol{x}=[2\quad 2\quad 1]^{\mathrm{T}}$$

程序(4.7) 正定对称矩阵平方根分解的 MATLAB 程序。

程序任务:对正定对称矩阵 $\boldsymbol{A}$ 进行$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 分解。

```
function L = MatSqrt(A)
% 输入:A—— 正定对称矩阵(N×N)
% 输出:L—— 下三角平方根矩阵(N×N)
n = length(A);              %n 是方阵 A 的行数和列数
L = zeros(n);               % 初始化方阵 L
L(1,1) = sqrt(A(1,1));      % 计算 L(1,1)
for i = 2:n
    L(i,1) = A(i,1)/L(1,1);  % 计算 L(i,1)
    if i > 2
```

```
        for j = 2:(i-1)
            L(i,j) = (A(i,j) - L(i,1:(j-1)) * L(j,1:(j-1))')/L(j,j);   % 计算 L(i,j)
        end
    end
    L(i,i) = sqrt(A(i,i) - L(i,1:(i-1)) * L(i,1:(i-1))');   % 计算 L(i,i)
end
```

用程序(4.7)来计算正定对称矩阵

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 10 & 6 & 7 & 8 \\ 3 & 6 & 20 & 10 & 11 \\ 4 & 7 & 10 & 30 & 12 \\ 5 & 8 & 11 & 12 & 40 \end{bmatrix}$$

的平方根分解矩阵 $\boldsymbol{L}$。

在 MATLAB 命令窗口输入：

```
>> A=[1,2,3,4,5; 2,10,6,7,8; 3,6,20,10,11; 4,7,10,30,12; 5,8,11,12,40];
                                    % 输入矩阵 A
>> L=MatSqrt(A)      % 调用程序(4.7)计算平方根分解矩阵 L
```

输出结果：

```
L =
    1.0000         0         0         0         0
    2.0000    2.4495         0         0         0
    3.0000         0    3.3166         0         0
    4.0000   -0.4082   -0.6030    3.6701         0
    5.0000   -0.8165   -1.2060   -2.4688    2.6046
>> L*L'          % 验证计算结果
ans =
      1.0000    2.0000    3.0000    4.0000    5.0000
      2.0000   10.0000    6.0000    7.0000    8.0000
      3.0000    6.0000   20.0000   10.0000   11.0000
      4.0000    7.0000   10.0000   30.0000   12.0000
      5.0000    8.0000   11.0000   12.0000   40.0000
```

程序(4.8)　用矩阵平方根分解法解线性方程组的 MATLAB 程序。

程序任务：对正定对称矩阵 $\boldsymbol{A}$ 进行 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$ 分解，并解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$。

```
function x = EqsSqrt(A,b)
% 输入:A—— 方程组的系数矩阵(N×N,正定对称矩阵)
%      b—— 方程组的常数向量(N×1)
% 输出:x—— 方程组的解向量(N×1)
% 以下程序段对正定对称矩阵 A 进行 A = LL^T 分解
n = length(A);                    %n 是方阵 A 的行数和列数
```

```
L = zeros(n);                    % 初始化方阵 L
L(1,1) = sqrt(A(1,1));           % 计算 L(1,1)
for i = 2:n
    L(i,1) = A(i,1)/L(1,1);      % 计算 L(i,1)
    if i > 2
        for j = 2:(i-1)
            L(i,j) = (A(i,j) - L(i,1:(j-1)) * L(j,1:(j-1))')/L(j,j);   % 计算 L(i,j)
        end
    end
    L(i,i) = sqrt(A(i,i) - L(i,1:(i-1)) * L(i,1:(i-1))');   % 计算 L(i,i)
end
% 以下程序段解方程组 Ly = b
y = zeros(n,1);          % 初始化向量 y
y(1) = b(1)/L(1,1);  % 计算 y(1)
for i = 2:n
    y(i) = (b(i) - L(i,1:(i-1)) * y(1:(i-1)))/L(i,i);      % 计算 y(2 ~ n)
end
% 以下程序段解方程组L^T x = y
x = zeros(n,1);          % 初始化向量 x
x(n) = y(n)/L(n,n);  % 计算 x(n)
for i = (n-1):-1:1
    x(i) = (y(i) - L((i+1):n ,i)' * x((i+1):n))/L(i,i);   % 计算 x((n-1) ~ 1)
end
```

用程序(4.8)解方程组

$$\begin{cases} 2x_1 + x_2 + x_3 = 0 \\ x_1 + x_2 + x_3 = 3 \\ x_1 + x_2 + 2x_3 = 1 \end{cases}$$

在 MATLAB 命令窗口输入：

```
>> A = [2,1,1; 1,1,1; 1,1,2];          % 输入系数矩阵
>> b = [0,3,1]';                       % 输入常数向量
>> x = EqsSqrt(A,b)                    % 调用程序(4.8)解方程
```

输出结果：

```
x = [-3.0000    8.0000    -2.0000]'
```

即方程组的解是 $\boldsymbol{x} = [-3.000\,0 \quad 8.000\,0 \quad -2.000\,0]'$。

为了克服平方根方法中的开平方运算，对平方根方法做进一步简化，形成了改进的平方根法。该方法解线性方程组的过程如下：

(1) 对系数矩阵进行分解，即

$$\boldsymbol{A} = \boldsymbol{L}_0 \boldsymbol{D} \boldsymbol{L}_0^{\mathrm{T}} \tag{4.37}$$

式中

$$L_0=\begin{bmatrix}1 & 0 & \cdots & 0\\ l_{21} & 1 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ l_{n1} & l_{n2} & \cdots & 1\end{bmatrix},\quad D=\begin{bmatrix}d_1 & 0 & \cdots & 0\\ 0 & d_2 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & d_n\end{bmatrix} \tag{4.38}$$

$$l_{ij}=\frac{a_{ij}-\sum_{r=1}^{j-1}l_{ir}d_r l_{jr}}{d_j}\quad (j=1,2,\cdots,i-1) \tag{4.39}$$

$$d_i=a_{ii}-\sum_{r=1}^{i-1}d_r l_{ir}^2\quad (i=1,2,\cdots,n) \tag{4.40}$$

(2) 令 $DL_0^{\mathrm T}x=y$，则有$L_0y=b$，$L_0^{\mathrm T}x=D^{-1}y$。解方程组$L_0y=b$，有

$$y_i=b_i-\sum_{r=1}^{i-1}l_{ir}y_r\quad \left(i=1,2,\cdots,n;\ \sum_{r=1}^{0}=0\right) \tag{4.41}$$

(3) 解方程组$L_0^{\mathrm T}x=D^{-1}y$，得

$$x_i=\frac{y_i}{d_i}-\sum_{r=i+1}^{n}l_{ri}x_r\quad \left(i=n,n-1,\cdots,1;\ \sum_{r=n+1}^{n}=0\right) \tag{4.42}$$

例 4.6　用改进的平方根方法解方程组

$$\begin{bmatrix}1 & 2 & 1\\ 2 & 8 & 4\\ 1 & 4 & 6\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}0\\ -2\\ 3\end{bmatrix}$$

解　由于

$$a_{11}=1>0,\quad \begin{vmatrix}1 & 2\\ 2 & 8\end{vmatrix}=4>0,\quad \begin{vmatrix}1 & 2 & 1\\ 2 & 8 & 4\\ 1 & 4 & 6\end{vmatrix}=16>0$$

因此方程组的系数矩阵是对称正定的。对其进行分解，即$A=L_0DL_0^{\mathrm T}$，得出单位下三角矩阵和对角矩阵分别是

$$L_0=\begin{bmatrix}1 & & \\ 2 & 1 & \\ 1 & \frac{1}{2} & 1\end{bmatrix},\quad D=\begin{bmatrix}1 & & \\ & 4 & \\ & & 4\end{bmatrix}$$

解方程组 $L_0y=b$，即

$$\begin{bmatrix}1 & & \\ 2 & 1 & \\ 1 & \frac{1}{2} & 1\end{bmatrix}\begin{bmatrix}y_1\\ y_2\\ y_3\end{bmatrix}=\begin{bmatrix}0\\ -2\\ 3\end{bmatrix}$$

有

$$y=[0\quad -2\quad 4]^{\mathrm T}$$

解方程组 $L_0^{\mathrm T}x=D^{-1}y$，即

$$\begin{bmatrix}1 & 2 & 1\\ & 1 & \frac{1}{2}\\ & & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}0\\ -\frac{1}{2}\\ 1\end{bmatrix}$$

最终求出原方程组的解

$$x = [1 \quad -1 \quad 1]^T$$

4. 追赶法

当方程组的系数矩阵 A 是下列形式的三对角矩阵时，即

$$A = \begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix} \tag{4.43}$$

（其中的空白部分均为零元素）称线性方程组

$$Ax = d \tag{4.44}$$

为三对角方程组。其中

$$d = [d_1 \quad d_2 \quad \cdots \quad d_n]^T \tag{4.45}$$

对系数矩阵 A 进行克洛特分解，即

$$A = L\tilde{U}_0 \tag{4.46}$$

式中

$$L = \begin{bmatrix} u_1 & & & & \\ w_2 & u_2 & & & \\ & \ddots & \ddots & & \\ & & w_{n-1} & u_{n-1} & \\ & & & w_n & u_n \end{bmatrix}, \quad \tilde{U}_0 = \begin{bmatrix} 1 & v_1 & & & \\ & 1 & v_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & v_{n-1} \\ & & & & 1 \end{bmatrix} \tag{4.47}$$

其中

$$\left.\begin{array}{ll} u_1 = b_1 & v_1 = \dfrac{c_1}{b_1} \\ w_i = a_i & (i = 2,3,\cdots,n) \\ u_i = b_i - w_i v_{i-1} & (i = 2,3,\cdots,n) \\ v_i = \dfrac{c_i}{u_i} & (i = 2,3,\cdots,n-1) \end{array}\right\} \tag{4.48}$$

解方程组的步骤如下：

（1）令 $\tilde{U}_0 x = y$，解方程组 $Ly = d$，得

$$y_1 = \frac{d_1}{b_1}, \quad y_i = \frac{d_i - w_i y_{i-1}}{u_i} \quad (i = 2,3,\cdots,n) \tag{4.49}$$

（2）解方程组 $\tilde{U}_0 x = y$，有

$$x_n = y_n, \quad x_i = y_i - v_i x_{i+1} \quad (i = n-1, n-2, \cdots, 1) \tag{4.50}$$

通常，形象地把下标从小到大计算 y 的过程称为“追”，将下标由大到小计算 x 的过程称为“赶”，所以此方法叫作追赶法。

定理(4.4)　在三对角矩阵 A 中，若 ①$a_i \neq 0 (i = 2,3,\cdots,n)$ 且 $c_i \neq 0 (i = 1,2,\cdots,n-1)$；②$|b_1| > |c_1| > 0$，$|b_i| > |a_i| + |c_i|$ $(i = 2,3,\cdots,n-1)$，$|b_n| > |a_n| > 0$，则 A 非奇异且唯一存在形如式(4.46)的分解形式。

例 4.7　用追赶法解三对角线性代数方程组

$$\begin{bmatrix} 2 & -1 & & \\ -1 & 3 & -2 & \\ & -2 & 4 & -3 \\ & & -3 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}=\begin{bmatrix} 6 \\ 1 \\ -2 \\ 1 \end{bmatrix}$$

解　对系数矩阵 $\boldsymbol{A}$ 进行克洛特分解 $\boldsymbol{A}=\boldsymbol{L}\widetilde{\boldsymbol{U}}_0$，有

$$\begin{bmatrix} 2 & -1 & & \\ -1 & 3 & -2 & \\ & -2 & 4 & -3 \\ & & -3 & 5 \end{bmatrix}=\begin{bmatrix} 2 & & & \\ -1 & \frac{5}{2} & & \\ & -2 & \frac{12}{5} & \\ & & -3 & \frac{5}{4} \end{bmatrix}\begin{bmatrix} 1 & -\frac{1}{2} & & \\ & 1 & -\frac{4}{5} & \\ & & 1 & -\frac{5}{4} \\ & & & 1 \end{bmatrix}$$

解方程组 $\boldsymbol{L}\boldsymbol{y}=\boldsymbol{d}$，即

$$\begin{bmatrix} 2 & & & \\ -1 & \frac{5}{2} & & \\ & -2 & \frac{12}{5} & \\ & & -3 & \frac{5}{4} \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}=\begin{bmatrix} 6 \\ 1 \\ -2 \\ 1 \end{bmatrix}$$

得

$$\boldsymbol{y}=\begin{bmatrix} 3 & \frac{8}{5} & \frac{1}{2} & 2 \end{bmatrix}^{\mathrm{T}}$$

再解方程组 $\widetilde{\boldsymbol{U}}_0\boldsymbol{x}=\boldsymbol{y}$，即

$$\begin{bmatrix} 1 & -\frac{1}{2} & & \\ & 1 & -\frac{4}{5} & \\ & & 1 & -\frac{5}{4} \\ & & & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}=\begin{bmatrix} 3 \\ \frac{8}{5} \\ \frac{1}{2} \\ 2 \end{bmatrix}$$

得到原方程组的解是

$$\boldsymbol{x}=[5 \quad 4 \quad 3 \quad 2]^{\mathrm{T}}$$

4.2　范数与方程组的状态

为了研究线性方程组的状态，分析方程组数值解的误差传递，讨论迭代法解线性方程组的收敛性，需要引入描述向量和矩阵“大小”的量，为此这里介绍向量和矩阵范数的概念，并讨论方程组的状态。

4.2.1 向量范数和矩阵范数

1. 向量范数

定义(4.1) 设向量 $\boldsymbol{x} \in \mathbf{R}^n$($n$ 维实向量空间),若 $\boldsymbol{x}$ 的某个实数值函数 $\|\boldsymbol{x}\|$ 满足下列条件:

(1) 非负性:$\|\boldsymbol{x}\| \geqslant 0$,且 $\|\boldsymbol{x}\| = 0$ 成立的充要条件是 $\boldsymbol{x} = \mathbf{0}$;

(2) 齐次性:$\|k\boldsymbol{x}\| = |k| \cdot \|\boldsymbol{x}\|$($k$ 为任意实数);

(3) 三角不等式:对于任意向量 $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{R}^n$,都有 $\|\boldsymbol{x}+\boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$。

则称函数 $\|\boldsymbol{x}\|$ 为 $\mathbf{R}^n$ 上向量 $\boldsymbol{x}$ 的范数。

向量范数是以向量为变量的非负函数,一个向量空间上可以定义多种不同范数。下面给出常用的向量"p -范数"的定义。

定义(4.2) 设向量 $\boldsymbol{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^{\mathrm{T}} \in \mathbf{R}^n$,则定义

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} \quad (1 \leqslant p \leqslant \infty) \tag{4.51}$$

为 $\mathbf{R}^n$ 上向量 $\boldsymbol{x}$ 的"p -范数"。当 p 分别取 2,∞ 或 1 时,就得到向量的"2 -范数""∞ -范数"和"1 -范数":

(1) 向量的"2 -范数"

$$\|\boldsymbol{x}\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2} \tag{4.52}$$

(2) 向量的"∞ -范数"

$$\|\boldsymbol{x}\|_\infty = \max_{1\leqslant i\leqslant n}\{|x_i|\} \tag{4.53}$$

(3) 向量的"1 -范数"

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i| \tag{4.54}$$

容易证明,这样定义的几种范数均满足向量范数的三个条件。为了简便,当不针对具体向量时,把范数符号中的向量省掉,将向量范数记作"$\|\cdot\|$",甚至写成"$\| \quad \|$"。下面的定理给出了向量 1 -范数、2 -范数和 ∞ -范数之间的关系。

定理(4.5)(范数的等价性) 对于任何向量 $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{R}^n$,有

(1) $\|\boldsymbol{x}\|_2 \leqslant \|\boldsymbol{x}\|_1 \leqslant \sqrt{n}\ \|\boldsymbol{x}\|_2$;

(2) $\|\boldsymbol{x}\|_\infty \leqslant \|\boldsymbol{x}\|_1 \leqslant \sqrt{n}\ \|\boldsymbol{x}\|_\infty$;

(3) $\|\boldsymbol{x}\|_\infty \leqslant \|\boldsymbol{x}\|_2 \leqslant \sqrt{n}\ \|\boldsymbol{x}\|_\infty$。

例 4.8 计算向量 $\boldsymbol{x} = [0 \quad 1 \quad 3 \quad -1]^{\mathrm{T}}$ 的 1 -范数、2 -范数和 ∞ -范数。

解 根据公式(4.54)、公式(4.52)和公式(4.53),向量 $\boldsymbol{x}$ 的 1 -范数、2 -范数和 ∞ -范数分别为

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i| = 0+1+3+1 = 5$$

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2} = \sqrt{0+1+9+1} = \sqrt{11}$$

$$\|\boldsymbol{x}\|_{\infty}=\max_{1\leqslant i\leqslant n}\{|x_i|\}=\max\{0\quad 1\quad 3\quad 1\}=3$$

定义(4.3)　在$\mathbf{R}^n$中的向量序列$\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(k)},\cdots$称为收敛于向量$\boldsymbol{x}$，当且仅当$\lim\limits_{k\to\infty}\|\boldsymbol{x}^{(k)}-\boldsymbol{x}\|=0$。

2. 矩阵的范数及其与向量范数的关系

定义(4.4)　设矩阵$\boldsymbol{A}\in\mathbf{R}^{n\times n}$，若$\boldsymbol{A}$的某个实数值函数$\|\boldsymbol{A}\|$满足下列条件：

(1) 非负性：$\|\boldsymbol{A}\|\geqslant 0$，且$\|\boldsymbol{A}\|=0$成立的充要条件是$\boldsymbol{A}=\boldsymbol{O}$；

(2) 齐次性：$\|k\boldsymbol{A}\|=|k|\cdot\|\boldsymbol{A}\|$　(k为任意实数)；

(3) 三角不等式：对于任意矩阵$\boldsymbol{A},\boldsymbol{B}\in\mathbf{R}^{n\times n}$，都有$\|\boldsymbol{A}+\boldsymbol{B}\|\leqslant\|\boldsymbol{A}\|+\|\boldsymbol{B}\|$；

(4) $\|\boldsymbol{AB}\|\leqslant\|\boldsymbol{A}\|\cdot\|\boldsymbol{B}\|$。

则称$\|\boldsymbol{A}\|$是$\mathbf{R}^{n\times n}$上矩阵$\boldsymbol{A}$的范数。

经常应用矩阵与向量的乘积，而且乘积得到的是向量，因而矩阵范数与向量范数应当有一定关系。

定义(4.5)　若给定的向量范数和矩阵范数，对于任意向量$\boldsymbol{x}\in\mathbf{R}^n$和任意矩阵$\boldsymbol{A}\in\mathbf{R}^{n\times n}$，都有不等式

$$\|\boldsymbol{Ax}\|\leqslant\|\boldsymbol{A}\|\cdot\|\boldsymbol{x}\| \tag{4.55}$$

成立，则称所给的向量范数和矩阵范数是相容的。

下面给出由向量范数诱导出的矩阵范数，这种矩阵范数与对应的向量范数相容。

定义(4.6)　设向量$\boldsymbol{x}\in\mathbf{R}^n$，矩阵$\boldsymbol{A}\in\mathbf{R}^{n\times n}$，若给定了向量的一种范数$\|\boldsymbol{x}\|_p$，则定义

$$\|\boldsymbol{A}\|_p=\max_{x\neq 0}\left\{\frac{\|\boldsymbol{Ax}\|_p}{\|\boldsymbol{x}\|_p}\right\}\quad(p=1,2,\infty) \tag{4.56}$$

为矩阵$\boldsymbol{A}$的范数，并称为矩阵$\boldsymbol{A}$的算子范数。

矩阵的算子范数满足方阵范数的所有要求，并称其为$\mathbf{R}^{n\times n}$上从属于向量范数$\|\cdot\|_p$的矩阵范数，或称为由向量范数$\|\cdot\|_p$导出的矩阵范数，还称为矩阵$\boldsymbol{A}$的“p-范数”。下面给出从属于向量1-范数、2-范数和∞-范数的方阵范数。

定理(4.6)　设矩阵$\boldsymbol{A}$是n阶方阵，即$\boldsymbol{A}=[a_{ij}]_{n\times n}$，则

(1) $$\|\boldsymbol{A}\|_1=\max_{1\leqslant j\leqslant n}\left\{\sum_{i=1}^{n}|a_{ij}|\right\} \tag{4.57}$$

(2) $$\|\boldsymbol{A}\|_2=(\text{方阵}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\text{的最大特征值})^{1/2}=\sqrt{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})} \tag{4.58}$$

(3) $$\|\boldsymbol{A}\|_{\infty}=\max_{1\leqslant i\leqslant n}\left\{\sum_{j=1}^{n}|a_{ij}|\right\} \tag{4.59}$$

显然，$\|\boldsymbol{A}\|_1$是方阵各列元素绝对值之和的最大值，称为矩阵的列范数；$\|\boldsymbol{A}\|_{\infty}$是方阵各行元素绝对值之和的最大值，称为行范数。

例 4.9　计算矩阵$\boldsymbol{A}=\begin{bmatrix}1 & -2\\ -3 & 4\end{bmatrix}$的三种$p$-范数。

解　按照公式(4.57)、公式(4.58)和公式(4.59)，有

$$\|\boldsymbol{A}\|_1=\max\{4\quad 6\}=6$$

$$\|\boldsymbol{A}\|_2=\sqrt{15+\sqrt{221}}\approx 5.46$$

$$\|\boldsymbol{A}\|_{\infty}=\max\{3\quad 7\}=7$$

3. 矩阵的谱半径及其与矩阵范数的关系

定义(4.7) 设 n 阶方阵 $\boldsymbol{A}$ 的特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$，则称

$$\rho(\mathbf{A})=\max_{1\leqslant i\leqslant n}\{|\lambda_i|\} \tag{4.60}$$

为矩阵 $\boldsymbol{A}$ 的谱半径。

下面的定理给出了矩阵谱半径与范数的关系。

定理(4.7)(特征值上界) 矩阵 $\boldsymbol{A}$ 的谱半径不超过其任何一种算子范数，即

$$\rho(\mathbf{A})\leqslant\|\mathbf{A}\|_p \tag{4.61}$$

证明 设 λ 是 $\boldsymbol{A}$ 的任意一特征值，$\boldsymbol{x}$ 是与之对应的特征向量，则有

$$\boldsymbol{Ax}=\lambda\boldsymbol{x}$$

利用范数的性质有

$$|\lambda|\cdot\|\boldsymbol{x}\|=\|\lambda\boldsymbol{x}\|=\|\boldsymbol{Ax}\|\leqslant\|\boldsymbol{A}\|\cdot\|\boldsymbol{x}\|$$

特征向量 $\boldsymbol{x}\neq\boldsymbol{0}$，所以可以推导出

$$|\lambda|\leqslant\|\boldsymbol{A}\|$$

由 λ 的任意性可知定理成立。

4.2.2 方程组的状态和条件数

数值方法求解线性方程组的精度受两方面因素的影响，一方面是采用的数值方法，另一方面是方程组本身。线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 完全由系数矩阵和常数向量确定。从理论上讲，若系数矩阵和常数向量给定并且方程组有解，则方程组的解就完全确定。但用数值方法解线性方程组时，方程组系数矩阵和常数向量的微小变化(或误差)，有时对方程组的解影响很小，有时则影响很大。因此有必要研究方程组系数矩阵和常数向量的扰动对方程组解的影响，从而判断方程组数值解法的稳定性。这就是方程组的解关于系数矩阵和常数向量的敏感性问题，这样的分析方法称为“扰动分析”。方程组的敏感性与系数矩阵密切有关，为此引入描述系数矩阵性态的数 —— 条件数。

若线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵和常数向量各有一微小扰动 $\delta\boldsymbol{A}$ 和 $\delta\boldsymbol{b}$，它们给方程组的解向量带来了扰动 $\delta\boldsymbol{x}$，则扰动方程组是

$$(\mathbf{A}+\delta\mathbf{A})(\boldsymbol{x}+\delta\boldsymbol{x})=\boldsymbol{b}+\delta\boldsymbol{b} \tag{4.62}$$

其中，$\delta\boldsymbol{A}$ 称为系数矩阵 $\boldsymbol{A}$ 的扰动矩阵，$\delta\boldsymbol{x}$ 和 $\delta\boldsymbol{b}$ 分别称为向量 $\boldsymbol{x}$ 和 $\boldsymbol{b}$ 的扰动向量。

假设 $\boldsymbol{A}$ 和 $(\boldsymbol{A}+\delta\boldsymbol{A})$ 均为非奇异矩阵，则原方程组和扰动方程组都有解。下面分别讨论扰动 $\delta\boldsymbol{b}$ 和 $\delta\boldsymbol{A}$ 对方程组解向量的影响。

1. $\delta\boldsymbol{A}=\boldsymbol{O}$ 的情况

此时方程组的系数矩阵是精确的，而常数向量有一扰动 $\delta\boldsymbol{b}$，这里研究扰动 $\delta\boldsymbol{b}$ 对方程组解造成的相对扰动 $\|\delta\boldsymbol{x}\|/\|\boldsymbol{x}\|$。

$\boldsymbol{x}+\delta\boldsymbol{x}$ 满足的扰动方程组是

$$\boldsymbol{A}(\boldsymbol{x}+\delta\boldsymbol{x})=\boldsymbol{b}+\delta\boldsymbol{b}$$

从中减去 $\boldsymbol{Ax}=\boldsymbol{b}$ 后得

$$\boldsymbol{A}\delta\boldsymbol{x}=\delta\boldsymbol{b}$$

则

$$\delta\boldsymbol{x}=\boldsymbol{A}^{-1}\delta\boldsymbol{b}$$

根据范数的相容性，有不等式

$$\|\delta \boldsymbol{x}\| \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{b}\|$$

再从方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 得到不等式

$$\|\boldsymbol{b}\| \leqslant \|\boldsymbol{A}\| \cdot \|\boldsymbol{x}\|$$

从而得出

$$\|\delta \boldsymbol{x}\| \cdot \|\boldsymbol{b}\| \leqslant \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{x}\| \cdot \|\delta \boldsymbol{b}\|$$

因为 $\boldsymbol{b}\neq\boldsymbol{0}$，$\|\boldsymbol{x}\|\neq 0$，从上式推导出

$$\frac{\|\delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\| \cdot \frac{\|\delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|} \tag{4.63}$$

此式给出了解向量精确度的一种估计方法，它表明解向量的相对扰动不大于方程组常数向量相对扰动的（$\|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\|$）倍。

2. $\delta \boldsymbol{b}=\boldsymbol{0}$ 的情况

这时方程组的常数向量无扰动，而系数矩阵有一扰动，扰动方程是

$$(\boldsymbol{A}+\delta \boldsymbol{A})(\boldsymbol{x}+\delta \boldsymbol{x})=\boldsymbol{b}$$

从中减去 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$，得

$$\delta \boldsymbol{A}(\boldsymbol{x}+\delta \boldsymbol{x})+\boldsymbol{A}\delta \boldsymbol{x}=\boldsymbol{0}$$

则

$$\delta \boldsymbol{x}=-\boldsymbol{A}^{-1}\delta \boldsymbol{A}(\boldsymbol{x}+\delta \boldsymbol{x})$$

相应地有

$$\|\delta \boldsymbol{x}\| \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{A}\| \cdot \|\boldsymbol{x}+\delta \boldsymbol{x}\|$$

由于$(\boldsymbol{A}+\delta \boldsymbol{A})$可逆，且 $\boldsymbol{b}\neq\boldsymbol{0}$，$\|\boldsymbol{x}+\delta \boldsymbol{x}\|\neq 0$，推导出

$$\frac{\|\delta \boldsymbol{x}\|}{\|\boldsymbol{x}+\delta \boldsymbol{x}\|} \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\| \cdot \frac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|} \tag{4.64}$$

同样，方程组系数矩阵的扰动给方程组解带来的相对扰动也不大于系数矩阵相对扰动的（$\|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\|$）倍。

公式(4.63)和公式(4.64)均表明，$\|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\|$ 反映了方程组的解对系数矩阵和常数向量扰动的敏感程度，它表征着方程组的性态。

定义(4.8)　设 $\boldsymbol{A}$ 为非奇异矩阵，称 $\|\boldsymbol{A}^{-1}\| \cdot \|\boldsymbol{A}\|$ 为矩阵 $\boldsymbol{A}$ 的条件数。记作

$$\mathrm{Cond}(\boldsymbol{A})=\|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| \tag{4.65}$$

条件数与选用的矩阵范数有关。当采用矩阵的算子范数时，由于 $\|\boldsymbol{A}\|_p \cdot \|\boldsymbol{A}^{-1}\|_p \geqslant \|\boldsymbol{A}\boldsymbol{A}^{-1}\|_p=\|\boldsymbol{I}\|_p=1$，所以条件数是一个不小于 1 的数。

显然，方程组系数矩阵的条件数小，系数矩阵和常数向量的扰动引起方程组解的相对变化小；系数矩阵的条件数大，系数矩阵和常数向量的扰动引起的方程组解的相对变化就可能大。当方程组的系数矩阵和常数向量同时存在扰动时，这个结论仍然成立。在一般情况下，当 $\|\boldsymbol{A}^{-1}\delta \boldsymbol{A}\| \leqslant \|\boldsymbol{A}^{-1}\| \cdot \|\delta \boldsymbol{A}\| < 1$ 时，有

$$\frac{\|\delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leqslant \frac{\mathrm{Cond}(\boldsymbol{A})}{1-\mathrm{Cond}(\boldsymbol{A})\dfrac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|}}\left(\frac{\|\delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|}+\frac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|}\right) \tag{4.66}$$

当方程组系数矩阵的条件数相对较小时，称这个方程组是良态的；反之，当方程组系数矩阵的条件数相对较大时，称方程组为病态的。同时也分别称该矩阵为良态矩阵或病态矩阵。

用数值稳定的方法解良态方程组时，必然得到较精确的解；同理，用数值稳定的方法解病态方程组时，结果的精度就可能很差。

例 4.10 求解线性代数方程组

$$\begin{cases} x_1 + 10^4 x_2 = 10^4 \\ x_1 + x_2 = 2 \end{cases}$$

解 方程组的精确解是 $x_1 = 1, x_2 = 1$。由于系数矩阵的条件数 $\mathrm{Cond}_\infty(\boldsymbol{A}) = \dfrac{(10^4+1)^2}{10^4-1} \approx 10^4$ 很大，方程组是严重病态方程组。采用顺序高斯消去法，得出的解是 $x_1 = 0, x_2 = 1$，严重失真。

在实际应用上，对于病态方程组，一般可以采用高精度数值计算，以减轻病态矩阵对解的影响。有时也选择一个适当的数乘以方程组中的某个方程，能够改变系数矩阵的条件数，使方程组解的精度有所改善。

本题中用 10^{-4} 乘以第一个方程，方程组的系数矩阵变为 $\boldsymbol{B} = \begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix}$，从而条件数是 $\mathrm{Cond}_\infty(\boldsymbol{B}) = \dfrac{4}{1-10^{-4}} \approx 4$，比原方程组的条件数大为减小。采用列主元素消去法，可以解得 $x_1 = 1, x_2 = 1$，显然是一个相对精确的解。

4.3 解线性方程组的迭代法

解方程组的迭代法就是用逐步逼近的方法寻找方程组精确解的近似值。基本过程如下：

(1) 先构造方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 的同解方程组

$$\boldsymbol{x} = \boldsymbol{Mx} + \boldsymbol{g} \tag{4.67}$$

(2) 再依据上式写出迭代格式(或迭代公式)

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{Mx}^{(k)} + \boldsymbol{g} \quad (k = 0, 1, 2, \cdots) \tag{4.68}$$

其中，$\boldsymbol{M}$ 被称为迭代矩阵，$\boldsymbol{g}$ 被称为右端向量。

(3) 然后选定初始向量 $\boldsymbol{x}^{(0)} \in \mathbf{R}^n$，用迭代公式(4.68)逐步迭代，建立迭代序列 $\{\boldsymbol{x}^{(k)}\}$。

(4) 若序列 $\{\boldsymbol{x}^{(k)}\}$ 收敛，则称迭代格式(4.68)是收敛的，迭代序列的极限 $\boldsymbol{x}^*$ 就是方程组 $\boldsymbol{Ax} = \boldsymbol{b}$ 的解；若序列 $\{\boldsymbol{x}^{(k)}\}$ 不收敛，则称迭代格式(4.68)是发散的。

通常用误差控制量 ε 来控制迭代过程。当迭代向量误差 $(\boldsymbol{x}^* - \boldsymbol{x}^{(k)})$ 的范数小于给定控制量时，即

$$\| \boldsymbol{x}^* - \boldsymbol{x}^{(k)} \| < \varepsilon \tag{4.69}$$

则停止迭代，并以 $\boldsymbol{x}^{(k)}$ 作为方程组的近似解，$\boldsymbol{x}^* \approx \boldsymbol{x}^{(k)}$。有时也用相邻两次迭代向量差 $(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})$ 的范数小于给定精度 ε' 来控制迭代过程。当

$$\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \| < \varepsilon'$$

时停止迭代，并以 $\boldsymbol{x}^{(k+1)}$ 作为方程组的近似解，即 $\boldsymbol{x}^* \approx \boldsymbol{x}^{(k+1)}$。

用迭代格式(4.68)逐步迭代求解方程组的方法称为迭代法，也称为一阶线性定常迭代法。迭代法较为适合于求解系数矩阵为高阶稀疏矩阵的方程组，计算程序比较简单，但迭代法的一个重要问题是收敛性和收敛速度。

定理(4.8)(迭代法基本定理)　迭代公式$\boldsymbol{x}^{(k+1)}=\boldsymbol{M}\boldsymbol{x}^{(k)}+\boldsymbol{g}(k=0,1,2,\cdots)$对任意初始向量$\boldsymbol{x}^{(0)}\in\mathbf{R}^n$与右端向量$\boldsymbol{g}$都收敛的充分必要条件为$\rho(\boldsymbol{M})<1$。

定理(4.9)　设一般迭代公式$\boldsymbol{x}^{(k+1)}=\boldsymbol{M}\boldsymbol{x}^{(k)}+\boldsymbol{g}(k=0,1,2,\cdots)$的迭代矩阵$\boldsymbol{M}$满足$\|\boldsymbol{M}\|_p=q<1$,则对于任意初始向量$\boldsymbol{x}^{(0)}\in\mathbf{R}^n$与右端向量$\boldsymbol{g}$,迭代法都收敛于精确解$\boldsymbol{x}^*$。并且有误差估计式

$$\|\boldsymbol{x}^*-\boldsymbol{x}^{(k)}\|_p\leqslant\frac{q}{1-q}\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|_p,\quad\|\boldsymbol{x}^*-\boldsymbol{x}^{(k)}\|_p\leqslant\frac{q^k}{1-q}\|\boldsymbol{x}^1-\boldsymbol{x}^{(0)}\|_p\tag{4.70}$$

证明　根据定理(4.8),当迭代矩阵$\boldsymbol{M}$满足$\|\boldsymbol{M}\|=q<1$时,对于任意初始向量$\boldsymbol{x}^{(0)}\in\mathbf{R}^n$与右端向量$\boldsymbol{g}$,用迭代公式$\boldsymbol{x}^{(k+1)}=\boldsymbol{M}\boldsymbol{x}^{(k)}+\boldsymbol{g}(k=0,1,2,\cdots)$建立的迭代序列$\{\boldsymbol{x}^{(k)}\}$均收敛于方程组$\boldsymbol{x}=\boldsymbol{M}\boldsymbol{x}+\boldsymbol{g}$精确解$\boldsymbol{x}^*$。

$$\begin{aligned}\|\boldsymbol{x}^*-\boldsymbol{x}^{(k)}\|&=\|\boldsymbol{x}^*-\boldsymbol{x}^{(k+1)}+\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\|\leqslant\|\boldsymbol{x}^*-\boldsymbol{x}^{(k+1)}\|+\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\|=\\&\|\boldsymbol{M}(\boldsymbol{x}^*-\boldsymbol{x}^{(k)})\|+\|\boldsymbol{M}(\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)})\|\leqslant\\&\|\boldsymbol{M}\|\cdot\|\boldsymbol{x}^*-\boldsymbol{x}^{(k)}\|+\|\boldsymbol{M}\|\cdot\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|\end{aligned}$$

当$\|\boldsymbol{M}\|<1$时,移项并整理就有

$$\|\boldsymbol{x}^*-\boldsymbol{x}^{(k)}\|\leqslant\frac{\|\boldsymbol{M}\|}{1-\|\boldsymbol{M}\|}\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|$$

应用算子范数$\|\boldsymbol{M}\|_p=q$,就得式(4.70)的第一式。另外

$$\begin{aligned}\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|&=\|\boldsymbol{M}(\boldsymbol{x}^{(k-1)}-\boldsymbol{x}^{(k-2)})\|\leqslant\|\boldsymbol{M}\|\cdot\|\boldsymbol{x}^{(k-1)}-\boldsymbol{x}^{(k-2)}\|\leqslant\\&\|\boldsymbol{M}\|^2\cdot\|\boldsymbol{x}^{(k-2)}-\boldsymbol{x}^{(k-3)}\|\leqslant\cdots\leqslant\|\boldsymbol{M}\|^{(k-1)}\cdot\|\boldsymbol{x}^1-\boldsymbol{x}^{(0)}\|\end{aligned}$$

代入式(4.70)的第一式,可得第二式。

由式(4.70)的第一式可知,当$\|\boldsymbol{M}\|_p$比1小很多时,可以用$\|\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)}\|$来近似解的误差向量的范数$\|\boldsymbol{x}^*-\boldsymbol{x}^{(k)}\|$,也就是能用前后相邻两次迭代向量的范数小于事先给定的某一精度作为终止迭代过程的条件。

还可以用式(4.70)的第二式来估计迭代的次数。若要求解向量的近似值$\boldsymbol{x}^{(k)}$满足$\|\boldsymbol{x}^*-\boldsymbol{x}^{(k)}\|\leqslant\varepsilon$,则由式(4.70)的第二式可得

$$q^k\leqslant\frac{1-q}{\|\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\|}\varepsilon$$

两边取对数就是

$$k\ln q\leqslant\ln\left[\frac{1-q}{\|\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\|}\varepsilon\right]$$

考虑到$q<1$,于是有

$$k\geqslant\left|\frac{\ln\left[\dfrac{1-q}{\|\boldsymbol{x}^{(1)}-\boldsymbol{x}^{(0)}\|}\varepsilon\right]}{\ln q}\right|\tag{4.71}$$

需要指出的是,上面定理(4.9)给出的迭代法收敛条件$\|\boldsymbol{M}\|_p<1$是充分条件,因此当$\|\boldsymbol{M}\|_p>1$时并不能肯定迭代序列$\{\boldsymbol{x}^{(k)}\}$发散。

4.3.1　雅可比迭代法

雅可比(Jacobi)迭代法是最简单的迭代法,故此也称为简单迭代法。

1. 迭代公式和迭代过程

设 n 元线性代数方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

的系数矩阵非奇异，并且对角元素 $a_{ii}(i=1,2,\cdots,n)$ 均不等于零，则其等价方程组是

$$\begin{cases} x_1 = \dfrac{1}{a_{11}}(b_1 - a_{12}x_2 - \cdots - a_{1n}x_n) \\ x_2 = \dfrac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 - \cdots - a_{2n}x_n) \\ \cdots\cdots \\ x_n = \dfrac{1}{a_{nn}}(b_n - a_{n1}x_1 - \cdots - a_{n,n-1}x_{n-1}) \end{cases}$$

建立迭代公式

$$\left.\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - \cdots - a_{1n}x_n^{(k)}) \\ x_2^{(k+1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)}) \\ &\cdots\cdots \\ x_n^{(k+1)} &= \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)}) \end{aligned}\quad (k=0,1,2,\cdots)\right\} \tag{4.72}$$

选取初始向量 $\boldsymbol{x}^{(0)} = \{x_1^{(0)} \quad x_2^{(0)} \quad \cdots \quad x_n^{(0)}\}^{\mathrm{T}}$，代入迭代公式得到 $\boldsymbol{x}^{(1)} = \{x_1^{(1)} \quad x_2^{(1)} \quad \cdots \quad x_n^{(1)}\}^{\mathrm{T}}$，再将 $\boldsymbol{x}^{(1)}$ 代入迭代公式得到 $\boldsymbol{x}^{(2)}$，……，依次迭代就能形成迭代序列 $\{\boldsymbol{x}^{(k)}\}$，迭代序列的极限就是线性方程组的精确解。也可以用有限次迭代的结果作为方程组的近似解。迭代公式(4.72)的向量分量形式是

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j^{(k)}\right) \quad (i=1,2,\cdots,n;k=0,1,\cdots) \tag{4.73}$$

按照迭代公式(4.72)进行迭代、建立向量序列 $\{\boldsymbol{x}^{(k)}\}$ 的方法称为解线性方程组的雅可比迭代法，简称为J-迭代法。该迭代法的迭代公式非常简单，每次只需计算矩阵和向量的乘积，计算机运算程序也不复杂。

2. 迭代矩阵

把线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 变形为

$$\boldsymbol{Dx} + (\boldsymbol{A}-\boldsymbol{D})\boldsymbol{x} = \boldsymbol{b}$$

式中，对角矩阵 $\boldsymbol{D}$ 是

$$\boldsymbol{D} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \tag{4.74}$$

矩阵 $(\boldsymbol{A}-\boldsymbol{D})$ 的对角元素均等于零。由于矩阵 $\boldsymbol{D}$ 非奇异，上面的矩阵方程又能够写成

$$\boldsymbol{x}=(\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{A})\boldsymbol{x}+\boldsymbol{D}^{-1}\boldsymbol{b} \tag{4.75}$$

与式(4.67)对比可知,雅可比迭代法的迭代矩阵为

$$\boldsymbol{M}_{\mathrm{J}}=\boldsymbol{I}-\boldsymbol{D}^{-1}\boldsymbol{A} \tag{4.76}$$

并且右端向量为

$$\boldsymbol{g}_{\mathrm{J}}=\boldsymbol{D}^{-1}\boldsymbol{b}$$

例 4.11　用雅可比迭代法求解线性方程组

$$\begin{bmatrix}10 & 3 & 1\\ 2 & -10 & 3\\ 1 & 3 & 10\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}=\begin{bmatrix}14\\ -5\\ 14\end{bmatrix}$$

解　方程组的精确解是$\boldsymbol{x}^*=[1\quad 1\quad 1]^{\mathrm{T}}$。

根据公式(4.72),方程租的迭代公式是

$$\begin{cases}x_1^{(k+1)}=\dfrac{1}{10}(14-3x_2^{(k)}-x_3^{(k)})\\ x_2^{(k+1)}=\dfrac{1}{-10}(-5-2x_1^{(k)}-3x_3^{(k)})\quad (k=0,1,2,\cdots)\\ x_3^{(k+1)}=\dfrac{1}{10}(14-x_1^{(k)}-3x_2^{(k)})\end{cases}$$

取$\boldsymbol{x}^{(0)}=[0\quad 0\quad 0]^{\mathrm{T}}$,迭代六次的计算结果见表 4.1。

表　4.1

k	1	2	3	4	5	6
$\boldsymbol{x}^{(k)}$	$\begin{bmatrix}1.4\\0.5\\1.4\end{bmatrix}$	$\begin{bmatrix}1.11\\1.20\\1.11\end{bmatrix}$	$\begin{bmatrix}0.929\\1.055\\0.929\end{bmatrix}$	$\begin{bmatrix}0.9906\\0.9645\\0.9906\end{bmatrix}$	$\begin{bmatrix}1.0116\\0.9953\\1.0116\end{bmatrix}$	$\begin{bmatrix}1.000251\\1.005795\\1.000251\end{bmatrix}$

由表可见,$\boldsymbol{x}^{(6)}$已经十分接近精确解。

程序(4.9)　用雅可比迭代法解线性方程组的 MATLAB 程序。

程序任务:用雅可比迭代法解线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$。

```
function [x ni] = EqsJaco(A,b,x0,ep,Nm)
% 输入:A—— 方程组的系数矩阵(N×N,非奇异)
%       b—— 方程组的常数向量(N×1)
%       x0—— 迭代初值向量(N×1)
%       ep—— 近似解的精度
%       Nm—— 迭代次数上限
% 输出:x—— 方程组的近似解向量(N×1)
%       ni—— 实际迭代次数
n = length(A);                %n 是方阵 A 的行数和列数
ni = 0;                       % 初始化 ni
x = x0; x0 = x + 2 * ep;      % 初始化 x,以进入下面循环
while max(abs(x - x0)) > ep & ni < Nm, ni = ni+1; x0 = x;
```

```
    for i = 1:n
        y(i) = b(i);
        for j = 1:n
            if j ~= i
                y(i) = y(i) - A(i,j) * x0(j);      % 计算 b_i - Σ_{j=1, j≠i}^{n} a_ij x_j^(k)
            end
        end
        y(i) = y(i)/A(i,i);                        % 计算 x_i^(k+1) = 1/a_ii (b_i - Σ_{j=1, j≠i}^{n} a_ij x_j^(k))
    end
    x = y';
end
```

用程序(4.9)解方程组

$$\begin{cases}4x_1+3x_2=24\\3x_1+4x_2-x_3=30\\-x_2+4x_3=-24\end{cases}$$

在 MATLAB 命令窗口输入：

```
>> A=[4,3,0; 3,4,-1; 0,-1,4];                   % 输入系数矩阵
>> b=[24,30,-24]';                               % 输入常数向量
>> [x ni]=EqsJaco(A,b,[0,0,0]',1e-4,70)          % 调用程序(4.9)解方程
```

输出结果：

```
x=[3.0000 4.0000 -5.0000]'
ni=49
```

即方程的解为 $\boldsymbol{x}=[3.0000\quad 4.0000\quad -5.0000]'$，迭代次数为 49 次。

4.3.2 高斯-赛德尔迭代法

赛德尔迭代法也称高斯-赛德尔(Gauss-Seidel)迭代法，简记为G-S方法，它是简单迭代法的改进。

1. 迭代公式和迭代过程

若迭代序列$\{\boldsymbol{x}^{(k)}\}$收敛，则当 k 足够大时，$\boldsymbol{x}^{(k+1)}$ 比$\boldsymbol{x}^{(k)}$ 更接近方程组的精确解$\boldsymbol{x}^*$，同样 $x_i^{(k+1)}$ 比 $x_i^{(k)}$ 更靠近方程组精确解向量的第 i 个分量 x_i^*。在雅可比迭代法中，计算$\boldsymbol{x}^{(k+1)}$ 的分量 $x_i^{(k+1)}$ $(i>1)$ 时，仍使用$\boldsymbol{x}^{(k)}$ 的所有分量，而对已经计算出且精度更高的向量$\boldsymbol{x}^{(k+1)}$ 的分量 $x_1^{(k+1)}$，$x_2^{(k+1)}$，…，$x_{i-1}^{(k+1)}$ 没有及时利用。显然，若用组合向量$[x_1^{(k+1)}\quad\cdots\quad x_{i-1}^{(k+1)}\quad x_i^{(k)}\quad\cdots\quad x_n^{(k)}]^{\mathrm{T}}$ 代入迭代公式计算 $x_i^{(k+1)}$，则迭代序列的收敛速度可望有所提高。这正是高斯-赛德尔迭代法的思路。

赛德尔迭代法迭代公式的分量形式是

$$\left.\begin{aligned}
x_1^{(k+1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k)} - \cdots - a_{1n}x_n^{(k)}) \\
x_2^{(k+1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)}) \\
&\cdots\cdots \\
x_n^{(k+1)} &= \frac{1}{a_{nn}}(b_n - a_{n1}x_1^{(k+1)} - \cdots - a_{n,n-1}x_{n-1}^{(k+1)})
\end{aligned}\quad (k=0,1,2,\cdots)\right\} \tag{4.77}$$

即

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}) = x_i^{(k)} + \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}) \tag{4.78}$$

即用向量$[x_1^{(k)} \quad x_2^{(k)} \quad \cdots \quad x_n^{(k)}]^{\mathrm{T}}$计算$x_1^{(k+1)}$，用向量$[x_1^{(k+1)} \quad x_2^{(k)} \quad \cdots \quad x_n^{(k)}]^{\mathrm{T}}$计算$x_2^{(k+1)}$，……，用向量$[x_1^{(k+1)} \quad \cdots \quad x_{i-1}^{(k+1)} \quad x_i^{(k)} \quad \cdots \quad x_n^{(k)}]^{\mathrm{T}}$计算$x_i^{(k+1)}$，这就是高斯-赛德尔迭代格式的思路。

2. 迭代矩阵

把方程组$\boldsymbol{Ax}=\boldsymbol{b}$的系数矩阵$\boldsymbol{A}$分解为三个矩阵之和，即

$$\boldsymbol{A} = \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{U}$$

其中

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

则高斯-赛德尔迭代公式的矩阵形式是

$$\boldsymbol{x}^{(k+1)} = -\boldsymbol{D}^{-1}\boldsymbol{L}\,\boldsymbol{x}^{(k+1)} - \boldsymbol{D}^{-1}\boldsymbol{U}\,\boldsymbol{x}^{(k)} + \boldsymbol{D}^{-1}\boldsymbol{b} \tag{4.79}$$

或者是

$$(\boldsymbol{D}+\boldsymbol{L})\boldsymbol{x}^{(k+1)} = -\boldsymbol{U}\,\boldsymbol{x}^{(k)} + \boldsymbol{b}$$

设$a_{ii}\,(i=1,2,\cdots,n)$均不等于零，则矩阵$(\boldsymbol{D}+\boldsymbol{L})$非奇异，从而有

$$\boldsymbol{x}^{(k+1)} = -(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{U}\,\boldsymbol{x}^{(k)} + (\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{b}$$

与式(4.67)比较可得，高斯-赛德尔迭代法的迭代矩阵是

$$\boldsymbol{M}_{\mathrm{G}} = -(\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{U} \tag{4.80}$$

并且右端向量为

$$\boldsymbol{g}_{\mathrm{G}} = (\boldsymbol{D}+\boldsymbol{L})^{-1}\boldsymbol{b}$$

例 4.12　用高斯-赛德尔迭代法求解线性方程组

$$\begin{bmatrix} 10 & 3 & 1 \\ 2 & -10 & 3 \\ 1 & 3 & 10 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 14 \\ -5 \\ 14 \end{bmatrix}$$

解　方程组的精确解是$\boldsymbol{x}^{*} = [1 \quad 1 \quad 1]^{\mathrm{T}}$。

根据公式(4.77)，方程租的迭代公式是

$$\begin{cases} x^{(k+1)} = \dfrac{1}{10}(14 - 3y^{(k)} - z^{(k)}) \\ y^{(k+1)} = \dfrac{1}{-10}(-5 - 2x^{(k+1)} - 3z^{(k)}) \quad (k=0,1,2,\cdots) \\ z^{(k+1)} = \dfrac{1}{10}(14 - x^{(k+1)} - 3y^{(k+1)}) \end{cases}$$

取$\boldsymbol{x}^{(0)} = [0 \quad 0 \quad 0]^{\mathrm{T}}$，迭代五次的计算结果见表 4.2。

表 4.2

k	1	2	3	4	5
$\boldsymbol{x}^{(k)}$	$\begin{bmatrix}1.400\\0.780\\1.026\end{bmatrix}$	$\begin{bmatrix}1.063\,4\\1.020\,5\\0.987\,5\end{bmatrix}$	$\begin{bmatrix}0.995\,1\\0.995\,3\\1.001\,9\end{bmatrix}$	$\begin{bmatrix}1.001\,2\\1.000\,8\\0.999\,6\end{bmatrix}$	$\begin{bmatrix}0.999\,8\\0.999\,8\\1.000\,1\end{bmatrix}$

由表可见，$\boldsymbol{x}^{(5)}$ 已经十分接近精确解，精度高于雅可比迭代法的六次迭代精度。

程序(4.10) 用高斯-赛德尔迭代法解线性方程组的 MATLAB 程序。

程序任务：用高斯-赛德尔迭代解线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$。

```
function [x ni] = EqsGauSei(A,b,x0,ep,Nm)
% 输入:A—— 方程组的系数矩阵(N×N,非奇异)
%      b—— 方程组的常数向量(N×1)
%      x0—— 迭代初值向量(N×1)
%      ep—— 近似解的精度
%      Nm—— 迭代次数上限
% 输出:x—— 方程组的近似解量(N×1)
%      ni—— 实际迭代次数
n = length(A);        %n 是方阵 A 的行数和列数
ni = 0;
for i = 1:Nm
    ni = ni+1;
    for j = 1:n
        if j == 1
            x(1) = (b(1) - A(1,(2:n)) * x0(2:n))/A(1,1);            % 计算 x1
        elseif j == n
            x(n) = (b(n) - A(n,1:(n-1)) * (x(1:(n-1)))')/A(n,n);  % 计算 xn
        else
            %x 和 x0 分别存放第 i 和第(i-1) 次迭代近似解
            x(j) = (b(j) - A(j,1:(j-1)) * x(1:(j-1)) - A(j,(j+1):n) * x0((j+1):n))/A(j,j);
                                                                     % 计算 xj
        end
    end
        if max(abs(x' - x0)) < ep, break; end
    x0 = x';
end
```

```
x = x';
```

用程序(4.10)解方程组

$$\begin{cases}4x_1 + 3x_2 = 24 \\ 3x_1 + 4x_2 - x_3 = 30 \\ -x_2 + 4x_3 = -24\end{cases}$$

在 MATLAB 命令窗口输入：

```
>> A=[4,3,0; 3,4,-1; 0,-1,4];                     % 输入系数矩阵
>> b=[24,30,-24]';                                % 输入常数向量
>> [x ni]=EqsGauSei(A,b,[0,0,0]',1e-6,100)        % 调用程序(4.10) 解方程
```

输出结果：

```
x=[3.0000  4.0000  -5.0000]'
ni=30
```

即方程的解为 $\boldsymbol{x}=[3.0000\quad 4.0000\quad -5.0000]'$，迭代次数为 30 次。

4.3.3 雅可比迭代法和高斯-赛德尔迭代法的收敛性

定义(4.9)　若矩阵 $\boldsymbol{A}$ 满足 $|a_{ii}| > \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\ (i=1,2,\cdots,n)$，即在矩阵的每一行中，对角线元素的绝对值大于其他元素绝对值的和，则称矩阵 $\boldsymbol{A}$ 为按行严格对角占优矩阵。若矩阵 $\boldsymbol{A}$ 满足 $|a_{jj}| > \sum\limits_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}|\ (j=1,2,\cdots,n)$，即在矩阵的每一列中，对角线元素的绝对值大于其他元素绝对值的和，则称矩阵 $\boldsymbol{A}$ 为按列严格对角占优矩阵。如无特别指明是按行还是按列严格对角占优时，只简称 $\boldsymbol{A}$ 为严格对角占优矩阵。

定理(4.10)　若方阵 $\boldsymbol{A}$ 是严格对角占优矩阵，则其为非奇异矩阵。

证明　用反证法证明。若 $\boldsymbol{A}$ 是按行严格对角占优矩阵，即

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \quad (i=1,2,\cdots,n)$$

假设 $\boldsymbol{A}$ 奇异，则齐次方程组 $\boldsymbol{Ax}=\boldsymbol{0}$ 有非零解 $\bar{\boldsymbol{x}}$，非零解 $\bar{\boldsymbol{x}}$ 必有分量 $|\bar{x}_k| = \max\limits_{1\leqslant i\leqslant n}\{|\bar{x}_i|\} \neq 0$。对于非零解 $\bar{\boldsymbol{x}}$，齐次方程组 $\boldsymbol{Ax}=\boldsymbol{0}$ 的第 k 个方程是

$$\sum_{j=1}^{n} a_{kj}\bar{x}_j = 0$$

即

$$a_{kk}\bar{x}_k = -\sum_{\substack{j=1 \\ j\neq k}}^{n} a_{kj}\bar{x}_j$$

则

$$|a_{kk}| \cdot |\bar{x}_k| = \left|\sum_{\substack{j=1 \\ j\neq k}}^{n} a_{kj}\bar{x}_j\right| \leqslant \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{kj}| \cdot |\bar{x}_j| \leqslant |\bar{x}_k| \cdot \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{kj}|$$

得出

$$|a_{kk}| \leqslant \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{kj}|$$

这与 $\boldsymbol{A}$ 按行严格对角占优矛盾，说明 $\boldsymbol{A}$ 非奇异。

定理(4.11) 若方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 是严格对角占优矩阵，则求解方程组的雅可比迭代法和高斯-赛德尔迭代法均收敛。

证明 这里只证明高斯-赛德尔迭代法收敛。设 $\boldsymbol{A}$ 是严格对角占优矩阵，研究迭代矩阵的特征方程

$$|\lambda \boldsymbol{I} - \boldsymbol{M}_{\mathrm{G}}| = 0$$

将式(4.80)代入，有

$$|\lambda \boldsymbol{I} - \boldsymbol{M}_{\mathrm{G}}| = |\lambda \boldsymbol{I} + (\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U}| = |(\boldsymbol{L}+\boldsymbol{D})^{-1}[\lambda(\boldsymbol{L}+\boldsymbol{D})+\boldsymbol{U}]| =$$
$$|(\boldsymbol{L}+\boldsymbol{D})^{-1}| \cdot |\lambda(\boldsymbol{L}+\boldsymbol{D})+\boldsymbol{U}| = 0$$

由于已知条件能够保证 $|(\boldsymbol{L}+\boldsymbol{D})^{-1}| = |(\boldsymbol{L}+\boldsymbol{D})|^{-1} \neq 0$，因此

$$|\lambda(\boldsymbol{L}+\boldsymbol{D})+\boldsymbol{U}| = 0$$

即

$$|\lambda(\boldsymbol{L}+\boldsymbol{D})+\boldsymbol{U}| = \begin{vmatrix} \lambda a_{11} & a_{12} & \cdots & a_{1n} \\ \lambda a_{21} & \lambda a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ \lambda a_{n1} & \lambda a_{n2} & \cdots & \lambda a_{nn} \end{vmatrix} = 0$$

假设 $|\lambda| \geqslant 1$，再不妨设 $\boldsymbol{A}$ 是按行严格占优的，即

$$|a_{ii}| > \sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^{n} |a_{ij}|$$

两边同乘 $|\lambda|$ 就有

$$|\lambda| \cdot |a_{ii}| > |\lambda| \left(\sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^{n} |a_{ij}|\right) \geqslant \left(\sum_{j=1}^{i-1} |\lambda a_{ij}| + \sum_{j=i+1}^{n} |a_{ij}|\right)$$

说明 $\lambda(\boldsymbol{L}+\boldsymbol{D})+\boldsymbol{U}$ 也是严格对角占优矩阵，是非奇异矩阵，这与前面的结论 $|\lambda(\boldsymbol{L}+\boldsymbol{D})+\boldsymbol{U}| = 0$ 矛盾，故假设 $|\lambda| \geqslant 1$ 错误，因此只有 $|\lambda| < 1$，亦即 $\rho(\boldsymbol{M}_{\mathrm{G}}) < 1$，故高斯-赛德尔迭代法收敛。

定理(4.12) (1) 若方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 为对称正定矩阵，则高斯-赛德尔迭代法收敛。

(2) 若方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 为对称正定矩阵，$2\boldsymbol{D}-\boldsymbol{A}$ 也为对称正定矩阵，则雅可比迭代法收敛；若系数矩阵 $\boldsymbol{A}$ 为对称正定矩阵，$2\boldsymbol{D}-\boldsymbol{A}$ 为非正定矩阵，则雅可比迭代法不收敛。

4.3.4 松弛迭代法

1. 基本思想

松弛迭代法是高斯-赛德尔迭代法的一种加速方法，其基本思想是将高斯-赛德尔迭代法得到的第 $k+1$ 次迭代向量 $\boldsymbol{x}^{(k+1)}$ 与第 k 次迭代向量 $\boldsymbol{x}^{(k)}$ 作加权平均。当加权因子(即松弛因子)ω 选取适当时，加速效果显著。

松弛迭代法的向量形式为

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \omega(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) \tag{4.81}$$

这可以理解为一种误差补偿，若补偿合适，则收敛速度会明显加快。反之，可能使收敛速度降低。

把式(4.78)代入式(4.81)，求解方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的松弛迭代法计算公式为

$$x_i^{(k+1)}=(1-\omega)x_i^{(k)}-\omega\left(\sum_{j=1}^{i-1}\frac{a_{ij}}{a_{ii}}x_j^{(k+1)}+\sum_{j=i+1}^{n}\frac{a_{ij}}{a_{ii}}x_j^{(k)}-\frac{b_i}{a_{ii}}\right) \tag{4.82}$$

2. 迭代矩阵

把高斯-赛德尔迭代公式(4.79)代入式(4.81)右端，可得

$$\boldsymbol{D}\boldsymbol{x}^{(k+1)}=\omega(-\boldsymbol{L}\boldsymbol{x}^{(k+1)}-\boldsymbol{U}\boldsymbol{x}^{(k)}+\boldsymbol{b})+(1-\omega)\boldsymbol{D}\boldsymbol{x}^{(k)}$$

整理有

$$\boldsymbol{x}^{(k+1)}=(\boldsymbol{D}+\omega\boldsymbol{L})^{-1}[\boldsymbol{D}-\omega(\boldsymbol{D}+\boldsymbol{U})]\boldsymbol{x}^{(k)}+\omega(\boldsymbol{D}+\omega\boldsymbol{L})^{-1}\boldsymbol{b}$$

其迭代矩阵是

$$\boldsymbol{M}_S=(\boldsymbol{D}+\omega\boldsymbol{L})^{-1}[\boldsymbol{D}-\omega(\boldsymbol{D}+\boldsymbol{U})] \tag{4.83}$$

3. 收敛条件

定理(4.13)　松弛迭代法收敛的必要条件是松弛因子 ω 满足不等式

$$0<\omega<2$$

定理(4.14)　若方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 为正定矩阵，则用 $0<\omega<2$ 的松弛迭代法求解必收敛。

定理(4.15)　若方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 为严格对角占优矩阵，则当 $0<\omega\leqslant 1$ 时松弛迭代法求解必收敛。

定理(4.16)　若方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的系数矩阵 $\boldsymbol{A}$ 为对称正定矩阵，且还是三对角阵，则松弛迭代法的最佳松弛因子是

$$\omega=\frac{2}{1+\sqrt{1-\rho^2(\boldsymbol{M}_J)}} \tag{4.84}$$

其中，$\rho(\boldsymbol{M}_J)$ 是解方程组的雅可比迭代法迭代矩阵 $\boldsymbol{M}_J$ 的谱半径。

当 $0<\omega<1$ 时，松弛迭代法也称为低松弛迭代法(SUR 法)；当 $1<\omega<2$ 时，则一般称为超松弛迭代法(SOR 法)；当 $\omega=1$ 时，即为高斯-赛德尔迭代法。

例 4.13　分别用高斯-赛德尔迭代法和松弛迭代法求下列方程组的近似解：

$$\begin{bmatrix}-4 & 1 & 1 & 1\\ 1 & -4 & 1 & 1\\ 1 & 1 & -4 & 1\\ 1 & 1 & 1 & -4\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

取初值 $\boldsymbol{x}^{(0)}=[0\quad 0\quad 0\quad 0]^{\mathrm{T}}$，要求精度到 10^{-5}。

解　方程组的精确解是 $\boldsymbol{x}^*=[-1\quad -1\quad -1\quad -1]^{\mathrm{T}}$。

高斯-赛德尔迭代公式是

$$\begin{cases}x_1^{(k+1)}=-\dfrac{1}{4}(1-x_2^{(k)}-x_3^{(k)}-x_4^{(k)})\\ x_2^{(k+1)}=-\dfrac{1}{4}(1-x_1^{(k+1)}-x_3^{(k)}-x_4^{(k)})\\ x_3^{(k+1)}=-\dfrac{1}{4}(1-x_1^{(k+1)}-x_2^{(k+1)}-x_4^{(k)})\\ x_4^{(k+1)}=-\dfrac{1}{4}(1-x_1^{(k+1)}-x_2^{(k+1)}-x_3^{(k+1)})\end{cases}\quad(k=0,1,2,\cdots)$$

计算得到 $x^{(43)} = [-1.00000 \quad -1.00000 \quad -1.00000 \quad -1.00000]^T$，满足要求。

松弛迭代公式是

$$\begin{cases} x_1^{(k+1)} = x_1^{(k)} - \dfrac{\omega}{4}(1 + 4x_1^{(k)} - x_2^{(k)} - x_3^{(k)} - x_4^{(k)}) \\ x_2^{(k+1)} = x_2^{(k)} - \dfrac{\omega}{4}(1 - x_1^{(k+1)} + 4x_2^{(k)} - x_3^{(k)} - x_4^{(k)}) \\ x_3^{(k+1)} = x_3^{(k)} - \dfrac{\omega}{4}(1 - x_1^{(k+1)} - x_2^{(k+1)} + 4x_3^{(k)} - x_4^{(k)}) \\ x_4^{(k+1)} = x_4^{(k)} - \dfrac{\omega}{4}(1 - x_1^{(k+1)} - x_2^{(k+1)} - x_3^{(k+1)} + 4x_4^{(k)}) \end{cases} \quad (k=0,1,2,\cdots)$$

取不同松弛因子 ω，求出的满足精度要求的近似解列入表 4.3 中。

表 4.3

ω	1.0	1.1	1.2	1.3	1.4	1.5
k	21	16	11	10	13	16
$x^{(k)}$	$\begin{bmatrix}-0.99999\\-0.99999\\-1.00000\\-1.00000\end{bmatrix}$	$\begin{bmatrix}-0.99999\\-1.00000\\-1.00000\\-1.00000\end{bmatrix}$	$\begin{bmatrix}-1.00000\\-1.00000\\-1.00000\\-1.00000\end{bmatrix}$	$\begin{bmatrix}-1.00000\\-1.00000\\-1.00000\\-1.00000\end{bmatrix}$	$\begin{bmatrix}-1.00000\\-1.00000\\-1.00000\\-1.00000\end{bmatrix}$	$\begin{bmatrix}-0.99999\\-1.00000\\-1.00000\\-1.00000\end{bmatrix}$

由表可见，不同松弛因子有不同加速效果。

程序(4.11) 用超松弛迭代法解线性方程组的 MATLAB 程序。

程序任务：用超松弛迭代法解线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$。

```
function [x ni] = EqsSor(A,b,om,x0,ep,Nm)
% 输入:A—— 方程组的系数矩阵(非奇异)
%       b—— 方程组的常数向量
%       om—— 松弛因子(一般取值在 1 ~ 2 之间)
%       x0—— 迭代初值向量
%       ep—— 近似解的精度
%       Nm—— 迭代次数上限
% 输出:x—— 方程组的近似解向量
%       ni—— 实际迭代次数
n = length(A);    %n 是方阵 A 的行数和列数
x = x0; x0 = x + 2 * ep;
ni = 0;
L = tril(A, -1); U = triu(A,1);
while norm(x0 - x,inf) > ep & ni < Nm, ni = ni + 1; x0 = x;
   for i = 1:n
        xt(i) = (b(i) - L(i,1:(i-1)) * x(1:(i-1)) - U(i, (i+1):n) * x0( (i+1):n))/A(i,i);
        x(i) = (1 - om) * x0(i) + om * xt(i);   % 此两句是公式(4.82)
   end
end
```

```
if ni == Nm warning('迭代次数已达到上限'); end
```

用程序(4.11)解方程组

$$\begin{cases} -4x_1 + x_2 + x_3 + x_4 = 1 \\ x_1 - 4x_2 + x_3 + x_4 = 1 \\ x_1 + x_2 - 4x_3 + x_4 = 1 \\ x_1 + x_2 + x_3 - 4x_4 = 1 \end{cases}$$

在 MATLAB 命令窗口输入：

```
>> A=[-4,1,1,1; 1,-4,1,1; 1,1,-4,1; 1,1,1,-4];  % 输入系数矩阵
>> b=[1,1,1,1]';                                  % 输入常数向量
>> [x ni]=EqsSor(A,b,1.3,[0,0,0,0]',1e-6,100)     % 调用程序(4.11)解方程
```

输出结果：

```
x=[-1.0000  -1.0000  -1.0000  -1.0000]'
ni=14
```

即方程的解为 $\boldsymbol{x}=[-1.0000\quad -1.0000\quad -1.0000\quad -1.0000]'$，当松弛因子为1.3时，迭代次数为 14 次。

4.4　物理学中的应用举例——直流单臂电桥分析

如图 4.3 所示的直流单臂电桥由电源、四个电阻和检流计组成，电源电动势为 E、内阻忽略不计，四个电阻的阻值分别是 $R_k(k=1,2,3,4)$，检流计的内阻是 R_g，各支路电流强度 $I_k(k=1,2,3,4)$ 与 I_g 的正方向如图中所示。根据基尔霍夫电压定律和电流定律，建立如下方程组：

$$\left.\begin{aligned} & I_1R_1 + I_3R_3 = E \\ & I_1R_1 + I_gR_g - I_2R_2 = 0 \\ & I_3R_3 - I_4R_4 - I_gR_g = 0 \\ & I_1 - I_3 - I_g = 0 \\ & I_2 - I_4 + I_g = 0 \end{aligned}\right\} \tag{4.85}$$

当其他量均已知时，解线性方程组，可以求出支路电流强度 $I_i(i=1,2,3,4)$ 与 I_g。

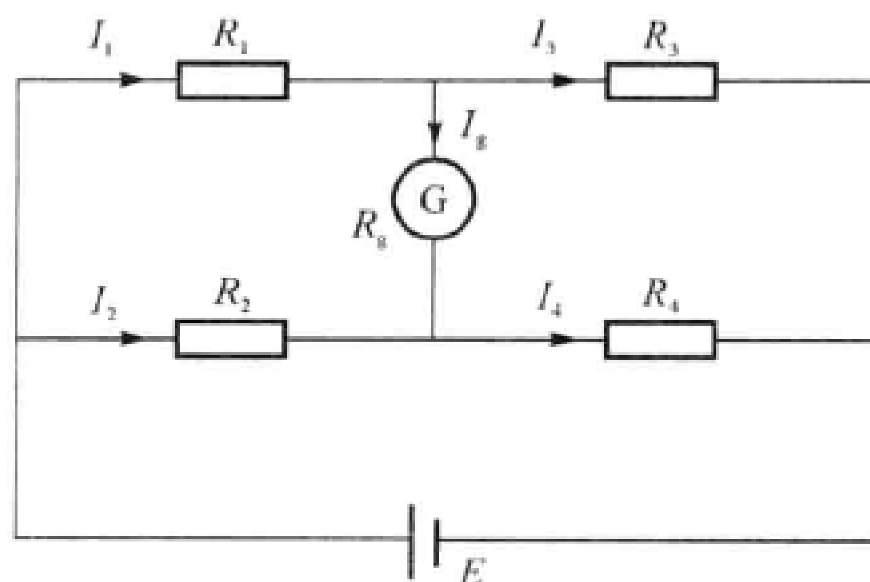


图　4.3

为了便于分析，利用后两个方程，从前三个方程中消去 I_3 和 I_4，得

$$\left.\begin{aligned}(R_1+R_3)I_1-R_gI_g&=E\\R_1I_1-R_2I_2+R_gI_g&=0\\R_3I_1-R_4I_2-(R_3+R_4+R_g)I_g&=0\end{aligned}\right\}\tag{4.86}$$

引入常量电流 $I_0=E/R_g$ 和无量纲电阻 $r_k=R_k/R_g(k=1,2,3,4)$，上式进一步写成

$$\left.\begin{aligned}r_3I_1-r_4I_2-(r_3+r_4+1)I_g&=0\\r_1I_1-r_2I_2+I_g&=0\\(r_1+r_3)I_1-I_g&=I_0\end{aligned}\right\}\tag{4.87}$$

用矩阵乘法表示为

$$\begin{bmatrix}r_3 & -r_4 & -(r_3+r_4+1)\\r_1 & -r_2 & 1\\(r_1+r_3) & 0 & -1\end{bmatrix}\begin{bmatrix}I_1\\I_2\\I_g\end{bmatrix}=I_0\begin{bmatrix}0\\0\\1\end{bmatrix}\tag{4.88}$$

为了分析电阻 $r_k(k=1,2,3,4)$ 的不同配置对电桥平衡特性的影响，分别选取几组不同的 $r_k(k=2,3,4)$，解线性方程组，计算 r_1 变化过程中，电流 I_1,I_2 与 I_g 随参数

$$\alpha=\frac{r_1/r_3}{r_2/r_4}\tag{4.89}$$

的变化情况。当 $\alpha=1$ 时，电桥平衡，$I_g=0$。

程序(4.12) 分析单臂直流电桥平衡特性的 MATLAB 程序。

程序任务：对于不同的 $r_k(k=2,3,4)$ 取值，当 r_1 变化时解线性方程组式(4.88)，并计算 α，绘制曲线 $I_g-\alpha$，$I_1-\alpha$ 和 $I_2-\alpha$。

```
% 分析单臂直流电桥特性的 MATLAB 程序。
r2 = 5;  r3 = 5;  r4 = 5;          % 给 r2 ~ r4 赋值
a0 = r2/r4;                        % 计算 r2/r4 值
k = 0;
for a1 = 6 * a0/10: a0/20:14 * a0/10;
    k = k+1;
    r1 = a1 * r3;                                                          % 计算 r1 值
    A(1,1) = r3;       A(1,2) = -r4;      A(1,3) = -(r3+ r4+1);            % 建立系数矩阵
    A(2,1) = r1;       A(2,2) = -r2;      A(2,3) = 1;
    A(3,1) = r1+ r3;   A(3,2) = 0;        A(3,3) = -1;
    b = [0 0 1]';                                                          % 建立常数矩阵
    x = coluGauss(A,b);                               % 列主元素高斯消去法(程序(4.2))解方程组
    I1(k) = x(1); I2(k) = x(2); Ig(k) = x(3);         % 取出 I1,I2 和 Ig 的值
    alfa(k) = (r1/r3)/(r2/r4);                        % 计算 α 值
end
ax0 = 0 * alfa;
subplot(1,3,1);                         % 绘制 Ig -α 曲线
plot(alfa,Ig,'-k',alfa,ax0,'--k');
xlabel('\alpha'); ylabel('I_g /I_0');
axis tight;
grid on;
gtext('r_2 = 5;  r_3 = 5;  r_4 = 5');   % 在图中标 r2,r3,r4 的值
```

```
subplot(1,3,2);                               % 绘制 I_1 - α 曲线
plot(alfa,I1,'-k');
xlabel('\alpha'); ylabel('I_1 /I_0');
axis tight;
grid on;
subplot(1,3,3);                               % 绘制 I_2 - α 曲线
plot(alfa,I2,'-k');
xlabel('\alpha'); ylabel('I_2 /I_0');
axis tight;
grid on;
```

对于不同的 $r_k(k=2,3,4)$ 取值，运行程序，绘制的曲线如图 4.4 所示。从图中总结出 α 在 $0.9\sim1.1$ 之间变化时，电流 I_g，I_1 和 I_2 的变化范围，列于表 4.4 中。

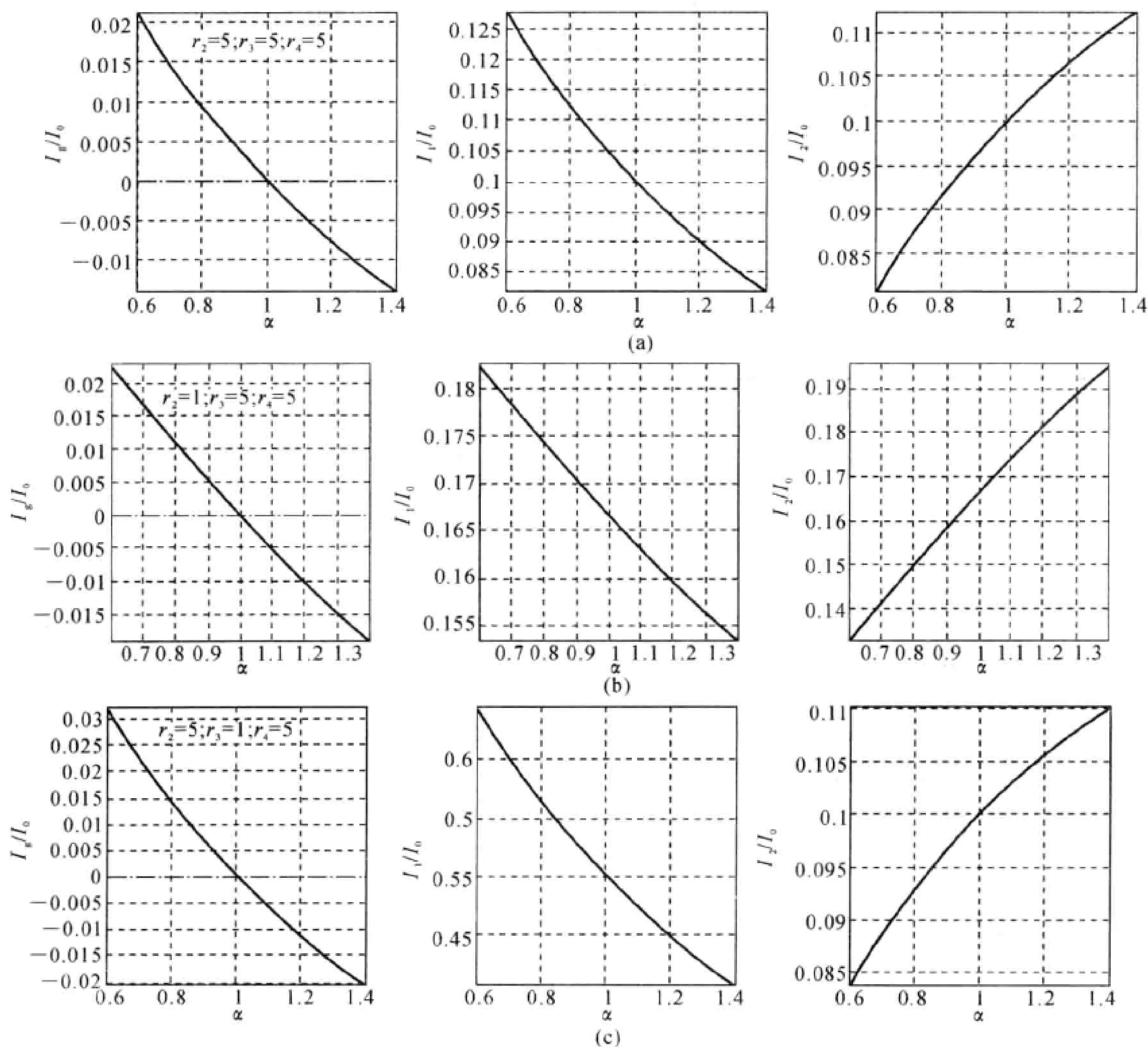
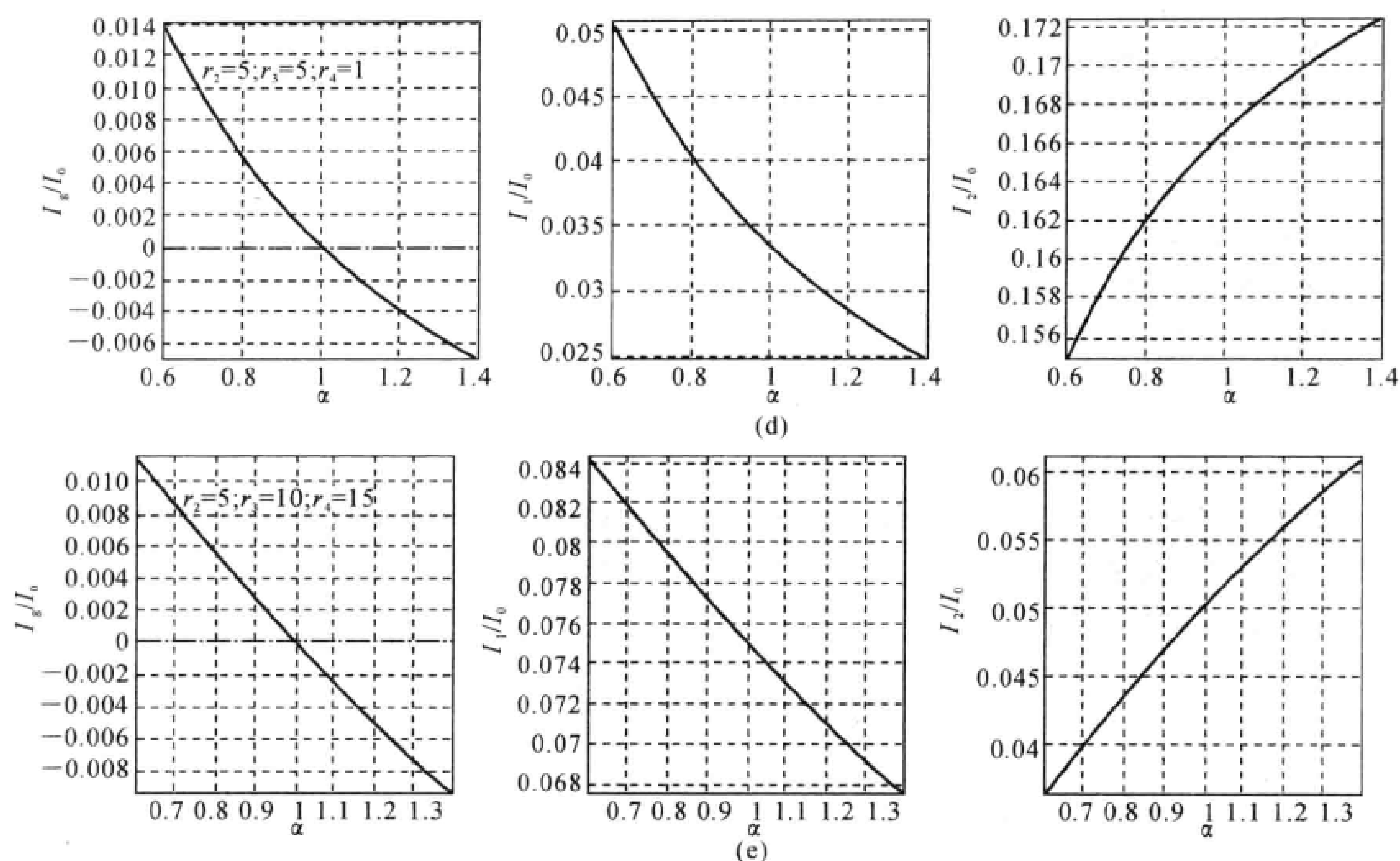


图　4.4

(a)$r_2=5, r_3=5, r_4=5$；　(b)$r_2=1, r_3=5, r_4=5$；　(c)$r_2=5, r_3=1, r_4=5$



续图 4.4

(d) $r_2=5, r_3=5, r_4=1$； (e) $r_2=5, r_3=10, r_4=15$

表 4.4

序号	$r_k(k=2,3,4)$ 取值	I_g 变化范围	I_1 变化范围	I_2 变化范围
1	$r_2=5, r_3=5, r_4=5$	$0.005\to-0.004$	$0.106\to0.095$	$0.096\to0.103$
2	$r_2=1, r_3=5, r_4=5$	$0.006\to-0.005$	$0.171\to0.167$	$0.158\to0.174$
3	$r_2=5, r_3=1, r_4=5$	$0.007\to-0.006$	$0.53\to0.47$	$0.097\to0.103$
4	$r_2=5, r_3=5, r_4=1$	$0.003\to-0.002$	$0.037\to0.031$	$0.165\to0.168$
5	$r_2=5, r_3=10, r_4=15$	$0.0026\to-0.0025$	$0.077\to0.073$	$0.047\to0.053$

从表中数据可以发现，在 α 从 0.9 到 1.1 的变化过程中，当 $r_2=5, r_3=1, r_4=5$ 时，I_g 的变化范围最大，说明电桥对平衡状态最为敏感。

习　题

1. 用顺序高斯消去法解方程组：

(1) $\begin{cases} 3x_1 - x_2 + 2x_3 = -3 \\ x_1 + x_2 + x_3 = -4 \\ 2x_1 + x_2 - x_3 = -3 \end{cases}$

(2) $\begin{cases} x_1 + 2x_2 + x_3 - 2x_4 = 4 \\ 2x_1 + 5x_2 + 3x_3 - 2x_4 = 7 \\ -2x_1 - 2x_2 + 3x_3 + 5x_4 = -1 \\ x_1 + 3x_2 + 2x_3 + 3x_4 = 0 \end{cases}$

2. 用列主元素高斯消去法解方程组：

(1) $\begin{cases} 2x_1+3x_2+5x_3=5 \\ 3x_1+4x_2+7x_3=6 \\ x_1+3x_2+3x_3=5 \end{cases}$　　(2) $\begin{cases} x_1+x_2-x_3=3 \\ 2x_1-x_2+3x_3=0 \\ -x_1-2x_2+x_3=-5 \end{cases}$

3. 分别用克洛特分解法和杜利特尔分解法对下列矩阵作 ***LU*** 分解。

(1) $\boldsymbol{A}=\begin{bmatrix} -2 & 4 & 8 \\ -4 & 18 & -16 \\ -6 & 2 & -20 \end{bmatrix}$　　(2) $\boldsymbol{B}=\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \\ 1 & 16 & 81 & 256 \end{bmatrix}$

4. 用克洛特分解法解方程组：

(1) $\begin{cases} 2x_1+6x_2-x_3=-12 \\ 5x_1-x_2+2x_3=29 \\ -3x_1-4x_2+x_3=5 \end{cases}$　　(2) $\begin{cases} x_1+2x_2+x_3=0 \\ 2x_1+2x_2+3x_3=3 \\ -x_1-3x_2=2 \end{cases}$

5. 用杜利特尔分解法解方程组：

(1) $\begin{cases} x_1-x_2+x_3=-4 \\ 5x_1-4x_2+3x_3=-12 \\ 2x_1+x_2+x_3=11 \end{cases}$　　(2) $\begin{cases} x_1+x_2-x_3=3 \\ 2x_1-x_2+3x_3=0 \\ -x_1-2x_2+x_3=-5 \end{cases}$

6. 分别用平方根方法和改进的平方根方法解方程组：

(1) $\begin{cases} 4x_1-2x_2-4x_3=10 \\ -2x_1+17x_2+10x_3=3 \\ -4x_1+10x_2+9x_3=7 \end{cases}$　　(2) $\begin{cases} 4x_1+2x_2+2x_3=3 \\ 2x_1+10x_2+x_3=6 \\ 2x_1+x_2+2x_3=2 \end{cases}$

7. 用追赶法解方程组：

(1) $\begin{cases} 5x_1+x_2=17 \\ x_1+5x_2+x_3=14 \\ x_2+5x_3=7 \end{cases}$　　(2) $\begin{cases} 2x_1-x_2=1 \\ -x_1+2x_2-x_3=\dfrac{1}{2} \\ -x_2+2x_3-x_4=\dfrac{1}{3} \\ -x_3+2x_4=\dfrac{1}{4} \end{cases}$

8. 设向量 $\boldsymbol{x}=[1\quad -4\quad 3]^{\mathrm{T}}$，矩阵 $\boldsymbol{A}=\begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix}$，求范数 $\|\boldsymbol{x}\|_1$，$\|\boldsymbol{x}\|_2$，$\|\boldsymbol{x}\|_\infty$ 及 $\|\boldsymbol{A}\|_1$，$\|\boldsymbol{A}\|_2$，$\|\boldsymbol{A}\|_\infty$。

9. 设矩阵 $\boldsymbol{A}=\begin{bmatrix} 1 & 0.99 \\ 0.99 & 0.98 \end{bmatrix}$，求其行范数条件数。

10. 分别用雅可比迭代法和高斯-赛德尔迭代法求解下列方程组。初始向量取为零向量，要求 $\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\|_\infty<\dfrac{1}{2}\times 10^{-3}$。

(1) $\begin{cases} 20x_1+2x_2+3x_3=24 \\ x_1+8x_2+x_3=12 \\ 2x_1-3x_2+15x_3=30 \end{cases}$　　(2) $\begin{cases} 5x_1-2x_2+x_3=4 \\ x_1+5x_2-3x_3=2 \\ 2x_1+x_2-5x_3=-11 \end{cases}$

11. 试对下列方程组进行等价变形后建立收敛的迭代公式，证明迭代公式是收敛的。并由此说明迭代法的收敛、发散性质可能由于方程组中方程或未知量的排列顺序的改变而改变。

(1) $\begin{cases} x_1 + 6x_2 - 2x_3 = 1 \\ 3x_1 - 2x_2 + 5x_3 = 2 \\ 4x_1 + x_2 - x_3 = 3 \end{cases}$　　(2) $\begin{cases} 3x_1 + x_2 + 2x_3 = 6 \\ 4x_2 + x_3 = 8 \\ x_1 + 2x_3 = 2 \end{cases}$

12. 试分别讨论用雅可比迭代法和高斯-赛德尔迭代法解下列方程组的收敛性。

(1) $\begin{cases} x_1 + 2x_2 - 2x_3 = 1 \\ x_1 + x_2 + x_3 = 2 \\ 2x_1 + 2x_2 + x_3 = -3 \end{cases}$　　(2) $\begin{cases} 2x_1 - x_2 + x_3 = 2 \\ x_1 + x_2 + x_3 = 1 \\ x_1 + x_2 - 2x_3 = 3 \end{cases}$

13. 用高斯-赛德尔迭代法解下列带状方程：

$$\begin{aligned}
12x_1 - 2x_2 + x_3 &= 5 \\
-2x_1 + 12x_2 - 2x_3 + x_4 &= 5 \\
x_1 - 2x_2 + 12x_3 - 2x_4 + x_5 &= 5 \\
x_2 - 2x_3 + 12x_4 - 2x_5 + x_6 &= 5 \\
&\cdots\cdots \\
x_{46} - 2x_{47} + 12x_{48} - 2x_{49} + x_{50} &= 5 \\
x_{47} - 2x_{48} + 12x_{49} - 2x_{50} &= 5 \\
x_{48} - 2x_{49} + 12x_{50} &= 5
\end{aligned}$$

14. 取$\boldsymbol{x}^{(0)} = [0 \quad 0 \quad 0]^{\mathrm{T}}$，用 SOR 法解下列方程组：

(1) $\begin{cases} 4x_1 + 3x_2 = 16 \\ 3x_1 + 4x_2 - x_3 = 20 \text{（取松弛因子 } \omega = 1.24\text{，要求 } \| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \|_\infty < \frac{1}{2} \times 10^{-4}\text{）} \\ -x_2 + 4x_3 = -12 \end{cases}$

(2) $\begin{cases} 5x_1 + 2x_2 + x_3 = -12 \\ -x_1 + 4x_2 + 2x_3 = 20 \text{（取松弛因子 } \omega = 0.9\text{，要求 } \| \boldsymbol{x}^{(k+1)} - x^{(k)} \|_\infty < \frac{1}{2} \times 10^{-4}\text{）} \\ 2x_1 - 3x_2 + 10x_3 = 3 \end{cases}$

15. 一圆锥曲线 $x^2 + bxy + cy^2 + dx + ey + f = 0$ 通过 5 个不同的点：$c_1(14.38, 3.94)$，$c_2(11.38, 2.79)$，$c_3(7.42, 3.07)$，$c_4(6.38, 5.11)$，$c_5(8.81, 2.59)$。建立圆锥曲线方程中的 5 个系数满足的线性方程组，并利用列主元素高斯消去法编程求解。

16. 在微积分中，下列积分可以通过部分分式技术求出：

$$\int \frac{x^2 + x + 1}{(x-1)(x-2)(x-3)^2(x^2+1)} \mathrm{d}x$$

这需要在下列表达式中计算出系数 $A_i (i = 1, 2, \cdots, 6)$：

$$\frac{x^2 + x + 1}{(x-1)(x-2)(x-3)^2(x^2+1)} = \frac{A_1}{(x-1)} + \frac{A_2}{(x-2)} + \frac{A_3}{(x-3)^2} + \frac{A_4}{(x-3)} + \frac{A_5 x + A_6}{(x^2+1)}$$

建立求解 $A_i (i=1,2,\cdots,6)$ 的线性代数方程组，编写解方程组的计算机程序并解出系数 $A_i (i=1,2,\cdots,6)$。

17. 基尔霍夫电压定律指出：电路网络中任意单向闭路的电压降（或升）的和为零。对图 4.5 中的电路利用基尔霍夫电压定律进行分析，可得到如下线性方程组：

$$\begin{cases}(R_1+R_3+R_4)I_1-R_3I_2-R_4I_3=E_1\\ R_3I_1-(R_2+R_3+R_5)I_2+R_5I_3=E_2\\ R_4I_1+R_5I_2-(R_4+R_5+R_6)I_3=0\end{cases}$$

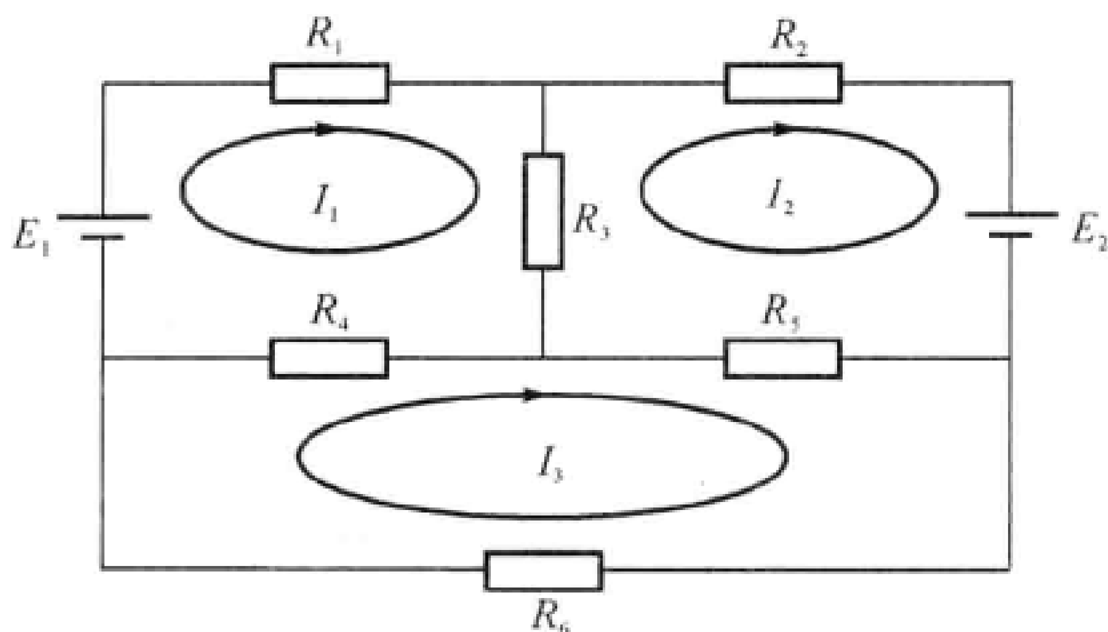


图　4.5

针对电阻和电源电动势的下列几组取值：

(1)$R_1=1\ \Omega,R_2=1\ \Omega,R_3=2\ \Omega,R_4=1\ \Omega,R_5=2\ \Omega,R_6=4\ \Omega,E_1=23\ \text{V},E_2=29\ \text{V}$；

(2)$R_1=1\ \Omega,R_2=0.75\ \Omega,R_3=1\ \Omega,R_4=2\ \Omega,R_5=1\ \Omega,R_6=4\ \Omega,E_1=12\ \text{V},E_2=21.5\ \text{V}$；

(3)$R_1=1\ \Omega,R_2=2\ \Omega,R_3=4\ \Omega,R_4=3\ \Omega,R_5=1\ \Omega,R_6=5\ \Omega,E_1=41\ \text{V},E_2=38\ \text{V}$。

分别选用解线性方程组的不同方法，编制计算机程序，求解电流强度 $I_i(i=1,2,3)$。

18. 在用有限元方法求解平面应力应变时，最后归结为下列线性代数方程组：

$$\begin{bmatrix}2.418 & -1.061 & 2.669 & 4.361 & -0.119 & -1.209 & -0.500\\ -1.501 & 19.832 & 0.694 & -4.816 & 2.274 & 2.001 & -1.909\\ 2.308 & 1.728 & -15.165 & -2.023 & 1.104 & 2.107 & -1.000\\ 3.359 & -0.913 & -6.441 & 27.864 & 3.737 & -4.375 & -2.375\\ -1.562 & 1.168 & -2.004 & 1.818 & 9.490 & 0.401 & -1.073\\ 1.174 & 7.318 & -2.278 & -0.143 & -9.835 & -31.670 & 4.114\\ 0.109 & -1.313 & -0.900 & -1.972 & -3.514 & -1.107 & 12.094\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\end{bmatrix}=\begin{bmatrix}8.262\\-33.818\\-52.673\\-97.284\\20.351\\149.9188\\81.653\end{bmatrix}$$

试用高斯-赛德尔迭代方法求解这个线性方程组。

19. 图 4.6 所示是一个交流电路，其中交流电源的电压按余弦规律变化，即 $V(t)=V_0\cos(\omega t)$，式中 V_0 是电压的振幅，ω 是电压的圆频率。根据交流电路知识，求解交流电路稳态解的方法与直流电路的求解方法类似，只要把各个元件的阻抗写成复阻抗（电阻 R 的复阻抗 $z_R=R$，电感 L 的复阻抗 $z_L=\mathrm{j}\omega L$，电容 C 的复阻抗 $z_C=1/(\mathrm{j}\omega C)$，这里 $\mathrm{j}=\sqrt{-1}$），把电源电压写成复数形式 $\tilde{V}(t)=V_0\mathrm{e}^{\mathrm{j}\omega t}$，列方程求解。然后，在解的表达式中取各个量的实部即可得到稳态解。用电位取代电压，并取 d 点电位为零。列出稳态解所满足的方程组

$$\begin{cases} V_b - 0 = i_2 R_2 \\ V_c - 0 = i_4(\mathrm{j}\omega L) \\ V_b - V_c = i_c/(\mathrm{j}\omega C) \\ V - V_b = i_1 R_1 \\ V - V_c = i_3 R_3 \\ i_1 = i_2 + i_c \\ i_4 = i_3 + i_c \end{cases}$$

编制根据已知条件 $V,\omega,L,C,R_k(k=1,2,3)$，解线性方程组计算 $V_b,V_c,i_c,i_k(k=1,2,3,4)$ 的计算机程序。

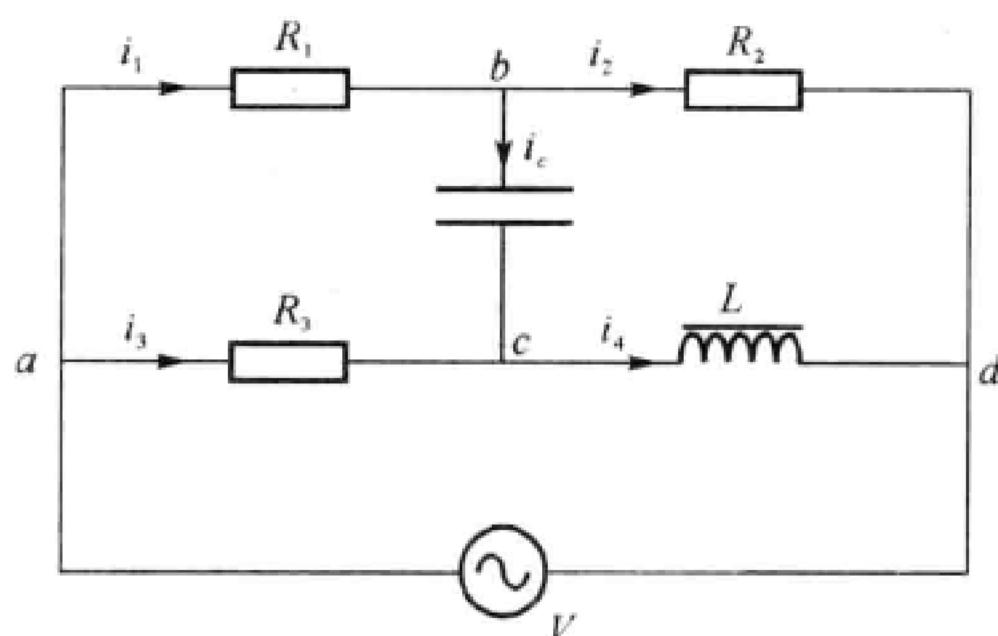


图 4.6

20. 如图 4.7 所示，质量为 m_1 的直角三角形物体放在水平面上，其斜边上有一质量为 m_2 的滑块，直角三角形物体的角 $\theta=\pi/6$，当地重力加速度 $g=10\ \mathrm{m/s^2}$，不计一切阻力。建立物体相对于水平面运动的加速度 a_1、滑块相对于物体运动的加速度 a_2、水平面对物体的支持力 N_1 和物体对滑块的支持力 N_2 所满足的线性方程组；若 m_1 与 m_2 选取以下几组值：

(1)$m_1=1\ \mathrm{kg}, m_2=5\ \mathrm{kg}$；

(2)$m_1=5\ \mathrm{kg}, m_2=5\ \mathrm{kg}$；

(3)$m_1=5\ \mathrm{kg}, m_2=1\ \mathrm{kg}$。

分别采用解线性方程组的不同方法，编制计算机程序，求 a_1,a_2,N_1 和 N_2 的值。

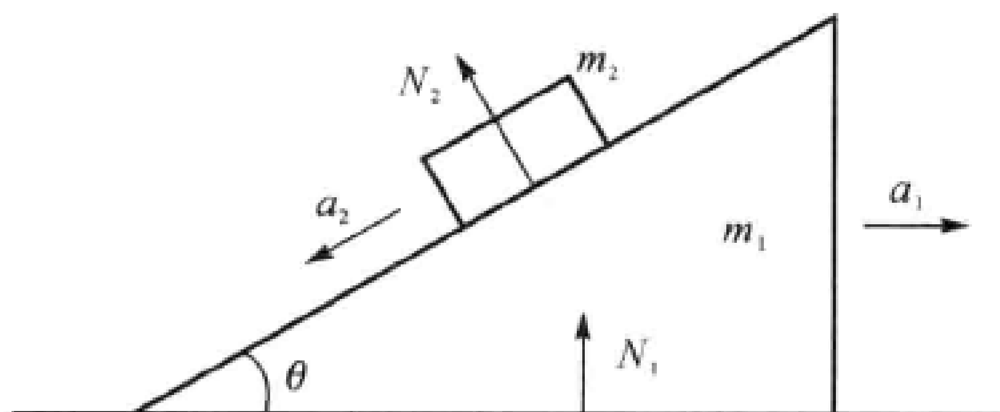


图 4.7

第5章 矩阵特征值与特征向量的计算

矩阵特征值和特征向量的求解是科学和工程计算中的一类重要问题，如动力学系统和结构系统的振动问题及物理学中的本征问题等都涉及系统特征值和特征向量的求解。本章介绍计算矩阵模最大特征值及对应特征向量的乘幂法、计算矩阵模最小特征值及对应特征向量的反幂法、计算实对称矩阵全部特征值及对应特征向量的雅可比方法。

5.1 矩阵的特征值和特征向量

定义(5.1) 对于方阵 $\boldsymbol{A}$，若有数 λ 及非零向量 $\boldsymbol{v}$ 满足

$$\boldsymbol{A}\boldsymbol{v}=\lambda\boldsymbol{v} \tag{5.1}$$

则称 λ 为矩阵 $\boldsymbol{A}$ 的特征值，$\boldsymbol{v}$ 为属于特征值 λ 的特征向量。

在线性代数中，计算矩阵的特征值和特征向量时，先求解特征方程

$$|\boldsymbol{A}-\lambda\boldsymbol{I}|=0 \tag{5.2}$$

即

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-\lambda \end{vmatrix}=0 \tag{5.3}$$

把上式中的行列式展开后，得到一个 n 次多项式

$$P(\lambda)=(-\lambda)^n+I_1(-\lambda)^{n-1}+I_2(-\lambda)^{n-2}+\cdots+I_{n-1}(-\lambda)+I_n \tag{5.4}$$

称为矩阵 $\boldsymbol{A}$ 的特征多项式，其中 $I_i(i=1,2,\cdots,n)$ 是矩阵 $\boldsymbol{A}$ 的主不变量。从特征方程解出矩阵的特征值 $\lambda_i(i=1,2,\cdots,n)$ 后，再解线性方程组

$$(\boldsymbol{A}-\lambda_i\boldsymbol{I})\boldsymbol{v}_i=\boldsymbol{0} \tag{5.5}$$

求解对应于特征值 λ_i 的特征向量 $\boldsymbol{v}_i$。

由于求解高次多项式方程式(5.3)本身是一件困难的事，因此很难得到高阶矩阵的特征值和特征向量。本章介绍三种简单而又有效的计算矩阵特征值和特征向量近似值的数值方法。

以下两个定理可用于判定矩阵特征值的存在性：

定理(5.1) 对于每一个唯一的特征值 λ，至少有一个与该特征值对应的特征向量 $\boldsymbol{v}$；如果特征值 λ 是特征方程式(5.3)的 m 重根，则至多有 m 个与该特征值相应的线性无关的特征向量。

定理(5.2) 设 $\lambda_i(i=1,2,\cdots,m)$ 是方阵 $\boldsymbol{A}$ 的互不相同的特征值，并且对应的特征向量为 $\boldsymbol{v}_i(i=1,2,\cdots,m)$，则 $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m\}$ 是一组线性无关的向量集合。

例 5.1 求下列矩阵的特征值和对应的特征向量：

$$A=\begin{bmatrix}3 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 3\end{bmatrix}$$

并证明这些特征向量线性无关。

解 矩阵 A 的特征方程为

$$|A-\lambda I|=\begin{vmatrix}3-\lambda & -1 & 0\\ -1 & 2-\lambda & -1\\ 0 & -1 & 3-\lambda\end{vmatrix}=-\lambda^3+8\lambda^2-19\lambda+12=-(\lambda-1)(\lambda-3)(\lambda-4)=0$$

得到三个特征值 $\lambda_1=1,\lambda_2=3,\lambda_3=4$。

将特征值 $\lambda_1=1$ 代入式(5.5),得到与特征值 λ_1 对应的特征向量 $\boldsymbol{v}_1=[x_{11}\quad x_{12}\quad x_{13}]^{\mathrm{T}}$ 满足的线性方程组

$$\begin{bmatrix}2 & -1 & 0\\ -1 & 1 & -1\\ 0 & -1 & 2\end{bmatrix}\begin{bmatrix}x_{11}\\ x_{12}\\ x_{13}\end{bmatrix}=\mathbf{0}$$

解出特征向量是 $\boldsymbol{v}_1=[a\quad 2a\quad a]^{\mathrm{T}}=a\,[1\quad 2\quad 1]^{\mathrm{T}}$,其中 a 是任意非零常数。采用同样的方法,求出与特征值 $\lambda_2=3$ 和 $\lambda_3=4$ 对应的特征向量分别是 $\boldsymbol{v}_2=b\,[1\quad 0\quad -1]^{\mathrm{T}}$ 和 $\boldsymbol{v}_3=c[1\quad -1\quad 1]^{\mathrm{T}}$,这里 b 和 c 也是任意非零常数。

由于三个特征值互不相同,根据定理(5.2)可知这些特征向量线性无关。另外,由于

$$|\boldsymbol{v}_1\quad \boldsymbol{v}_2\quad \boldsymbol{v}_3|=\begin{vmatrix}a & b & c\\ 2a & 0 & -c\\ a & -b & c\end{vmatrix}=-6abc\neq 0$$

因此特征向量 $\boldsymbol{v}_i\,(i=1,2,3)$ 线性无关。

将对角矩阵

$$D=\operatorname{diag}(\lambda_1,\lambda_2,\cdots,\lambda_n)=\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & \lambda_n\end{bmatrix}$$

与第 i 行元素等于1、其余元素均等于0的向量 $\boldsymbol{b}_i=[0\quad 0\quad \cdots\quad 0\quad 1\quad 0\quad \cdots\quad 0]^{\mathrm{T}}$ 相乘是

$$D\boldsymbol{b}_i=[0\quad 0\quad \cdots\quad 0\quad \lambda_i\quad 0\quad \cdots\quad 0]^{\mathrm{T}}=\lambda_i\boldsymbol{b}_i$$

与式(5.1)对比发现,向量 $\boldsymbol{b}_i$ 是矩阵 D 对应于特征值 λ_i 的特征向量。

这个分析表明,对角矩阵的特征值与特征向量是显而易见的。如果能把一个矩阵的特征值和特征向量与一个对角矩阵的特征值和特征向量联系起来,则会为求解矩阵的特征值和特征向量提供极大便利。两个相似矩阵的特征值和特征向量之间有着紧密联系。

定义(5.2) 若存在一个非奇异矩阵 K,使 $n\times n$ 矩阵 A 和 B 满足关系

$$B=K^{-1}AK \tag{5.6}$$

则称矩阵 A 与 B 是相似的。

定理(5.3) 设 A 与 B 是相似矩阵,$\boldsymbol{v}$ 为矩阵 A 对应于特征值 λ 的特征向量,那么 λ 也是矩阵 B 的特征值。如果 $K^{-1}AK=B$,则 $\boldsymbol{u}=K^{-1}\boldsymbol{v}$ 是 B 对应于特征值 λ 的特征向量。

定义(5.3) 若一个方阵与一个对角矩阵相似,则称这个方阵是可对角化的。

定理(5.4)(对角化定理)　$n\times n$ 矩阵 $\boldsymbol{A}$ 与对角矩阵 $\boldsymbol{D}$ 相似，当且仅当它有 n 个线性无关的特征向量。如果 $\boldsymbol{A}$ 与 $\boldsymbol{D}$ 相似，则有

$$\boldsymbol{V}^{-1}\boldsymbol{AV}=\boldsymbol{D}=\mathrm{diag}(\lambda_1,\quad \lambda_2,\quad \cdots,\quad \lambda_n) \tag{5.7}$$

式中

$$\boldsymbol{V}=[\boldsymbol{v}_1\quad \boldsymbol{v}_2\quad \cdots\quad \boldsymbol{v}_n] \tag{5.8}$$

其中，$\lambda_i(i=1,2,\cdots,n)$ 是矩阵 $\boldsymbol{A}$ 的特征值，$\boldsymbol{v}_i(i=1,2,\cdots,n)$ 是对应的特征向量。

定理(5.4)说明，具有 n 个不同特征值的 $n\times n$ 矩阵 $\boldsymbol{A}$ 是可对角化的。

5.2　乘　幂　法

矩阵的模最大的特征值称为主特征值。在许多实际问题中，主特征值起着重要作用，如迭代矩阵的谱半径决定着求解线性代数方程组迭代公式的收敛性。乘幂法就是用来计算大型稀疏矩阵主特征值及其对应特征向量的简单数值计算方法。

1. 计算方法

定义(5.4)　对于方阵 $\boldsymbol{A}$，任取初始向量 $\boldsymbol{x}^{(0)}$，利用递推公式

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{A}\boldsymbol{x}^{(k)}\quad (k=0,1,2,\cdots) \tag{5.9}$$

构造乘幂序列 $\{\boldsymbol{x}^{(k)}\}$；分析向量序列中向量 $\boldsymbol{x}^{(k)}$ 之间的关系，就可以得到矩阵 $\boldsymbol{A}$ 的主特征值和对应的特征向量。这种方法称为乘幂法。

2. 乘幂法分析

假设矩阵 $\boldsymbol{A}$ 有特征值 $\lambda_i(i=1,2,\cdots,n)$，并且 $|\lambda_1|\geqslant|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$，它们对应的 n 个线性无关特征向量是 $\boldsymbol{v}_i(i=1,2,\cdots,n)$。任取初始向量 $\boldsymbol{x}^{(0)}$，把它表示为向量 $\boldsymbol{v}_i(i=1,2,\cdots,n)$ 的线性组合，即

$$\boldsymbol{x}^{(0)}=\sum_{i=1}^{n}a_i\boldsymbol{v}_i \tag{5.10}$$

那么

$$\boldsymbol{x}^{(1)}=\boldsymbol{A}\boldsymbol{x}^{(0)}=\boldsymbol{A}\Big(\sum_{i=1}^{n}a_i\boldsymbol{v}_i\Big)=\sum_{i=1}^{n}a_i(\boldsymbol{A}\boldsymbol{v}_i)=\sum_{i=1}^{n}a_i\lambda_i\boldsymbol{v}_i$$

一般地，有

$$\boldsymbol{x}^{(k)}=\boldsymbol{A}\boldsymbol{x}^{(k-1)}=\sum_{i=1}^{n}a_i\lambda_i^k\boldsymbol{v}_i\quad (k=1,2,\cdots) \tag{5.11}$$

讨论几种特殊情况：

(1) 矩阵 $\boldsymbol{A}$ 有唯一的主特征值。此时，$|\lambda_1|>|\lambda_2|\geqslant\cdots\geqslant|\lambda_n|$，式(5.11)改写为

$$\boldsymbol{x}^{(k)}=\lambda_1^k\left[a_1\boldsymbol{v}_1+\sum_{i=2}^{n}a_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{v}_i\right] \tag{5.12}$$

若 $a_1\neq 0$，由于 $\left|\dfrac{\lambda_i}{\lambda_1}\right|<1(i=2,3,\cdots,n)$，当 k 充分大时，有

$$\boldsymbol{x}^{(k)}\approx\lambda_1^k a_1\boldsymbol{v}_1$$

$$\boldsymbol{x}^{(k+1)}\approx\lambda_1^{k+1}a_1\boldsymbol{v}_1\approx\lambda_1\boldsymbol{x}^{(k)}$$

写成向量分量关系就是

$$x_i^{(k+1)} \approx \lambda_1 x_i^{(k)} \quad (i=1,2,\cdots,n)$$

于是矩阵的主特征值是

$$\lambda_1 \approx x_i^{(k+1)}/x_i^{(k)} \quad (i=1,2,\cdots,n) \tag{5.13}$$

根据$\boldsymbol{x}^{(k+1)}=\boldsymbol{A}\boldsymbol{x}^{(k)}=\lambda_1\boldsymbol{x}^{(k)}$，与$\lambda_1$对应的特征向量的近似值为

$$\boldsymbol{v}_1 \approx \boldsymbol{x}^{(k)} \tag{5.14}$$

该方法中，$\{\boldsymbol{x}^{(k)}\}$的收敛速度取决于$|\lambda_2/\lambda_1|$的大小。

(2) 矩阵$\boldsymbol{A}$有两个相同的主特征值。设$\lambda_1=\lambda_2$为特征方程的二重根，则$|\lambda_1|=|\lambda_2|>|\lambda_3|\geqslant\cdots\geqslant|\lambda_n|$，有

$$\boldsymbol{x}^{(k)}=\lambda_1^k\left[a_1\boldsymbol{v}_1+a_2\boldsymbol{v}_2+\sum_{i=3}^{n}a_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{v}_i\right] \tag{5.15}$$

若$a_1\neq0,a_2\neq0$，由于$\left|\frac{\lambda_i}{\lambda_1}\right|<1(i=3,4,\cdots,n)$，当$k$充分大时，有

$$\boldsymbol{x}^{(k)} \approx \lambda_1^k(a_1\boldsymbol{v}_1+a_2\boldsymbol{v}_2)$$
$$\boldsymbol{x}^{(k+1)} \approx \lambda_1^{k+1}(a_1\boldsymbol{v}_1+a_2\boldsymbol{v}_2)$$

得到主特征值为

$$\lambda_1 \approx x_i^{(k+1)}/x_i^{(k)} \quad (i=1,2,\cdots,n) \tag{5.16}$$

对应于特征值λ_1,λ_2的特征向量的近似值为

$$\boldsymbol{v}_{1,2} \approx \boldsymbol{x}^{(k)} \tag{5.17}$$

二重主特征值的这种处理方法也可以推广到多重主特征值的情况。

(3) 矩阵$\boldsymbol{A}$的两个主特征值互为相反数。设$\lambda_1=-\lambda_2$，则$|\lambda_1|=|\lambda_2|>|\lambda_3|\geqslant\cdots\geqslant|\lambda_n|$，有

$$\boldsymbol{x}^{(k)}=\lambda_1^k\left[a_1\boldsymbol{v}_1+(-1)^k a_2\boldsymbol{v}_2+\sum_{i=3}^{n}a_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{v}_i\right] \tag{5.18}$$

若$a_1\neq0,a_2\neq0$，由于$\left|\frac{\lambda_i}{\lambda_1}\right|<1(i=3,4,\cdots,n)$，当$k$充分大时，有

$$\boldsymbol{x}^{(k)} \approx \lambda_1^k(a_1\boldsymbol{v}_1+(-1)^k a_2\boldsymbol{v}_2)$$
$$\boldsymbol{x}^{(k+1)} \approx \lambda_1^{k+1}(a_1\boldsymbol{v}_1+(-1)^{k+1}a_2\boldsymbol{v}_2)$$
$$\boldsymbol{x}^{(k+2)} \approx \lambda_1^{k+2}(a_1\boldsymbol{v}_1+(-1)^{k+2}a_2\boldsymbol{v}_2)=\lambda_1^2\boldsymbol{x}^{(k)}$$

得到主特征值为

$$\lambda_{1,2} \approx \pm\sqrt{x_i^{(k+2)}/x_i^{(k)}} \quad (i=1,2,\cdots,n) \tag{5.19}$$

对应于特征值λ_1,λ_2的特征向量的近似值分别为

$$\boldsymbol{v}_{1,2} \approx \boldsymbol{x}^{(k+1)} \pm \lambda_1\boldsymbol{x}^{(k)} \tag{5.20}$$

总之，计算出迭代序列$\{\boldsymbol{x}^{(k)}\}$后，当k充分大时，若比值$x_i^{(k+1)}/x_i^{(k)}$趋于一稳定值，则属于第(1)或第(2)种情况；如果比值$x_i^{(k+1)}/x_i^{(k)}$的符号交替变化，但绝对值趋于一稳定值，则属于第(3)种情况。

例 5.2 用乘幂法计算矩阵$\boldsymbol{A}=\begin{bmatrix}0&1\\1&1\end{bmatrix}$的模最大的特征值及对应的特征向量。

解 取初始向量为$\boldsymbol{x}^{(0)}=[1\quad 1]^{\mathrm{T}}$，用递推公式$\boldsymbol{x}^{(k+1)}=\boldsymbol{A}\boldsymbol{x}^{(k)}(k=0,1,2,\cdots)$计算，并进行分析，结果列入表 5.1 中。

表　5.1

k	$(x^{(k)})^{\mathrm{T}}$		$x_2^{(k)}/x_2^{(k-1)}$	k	$(x^{(k)})^{\mathrm{T}}$		$x_2^{(k)}/x_2^{(k-1)}$
1	1	2	2	7	21	34	1.619 05
2	2	3	1.5	8	34	55	1.617 65
3	3	5	1.666 67	9	55	89	1.618 18
4	5	8	1.6	10	89	144	1.61798
5	8	13	1.625	11	144	233	1.618 06
6	13	21	1.615	12	233	377	1.618 03

由表可以得出 $\lambda \approx 1.61803$，对应的特征向量是 $\boldsymbol{v}=[233\quad 377]^{\mathrm{T}}$。

3. *规范化的乘幂法公式*

在用乘幂法计算时，迭代向量的分量可能会过大，造成计算机溢出停机。为了避免这种现象的出现，可以采用规范化的乘幂法公式。

设

$$\max(\boldsymbol{x})=\max_{1\leqslant i\leqslant n}\{|x_i|\} \tag{5.21}$$

对于向量 $\boldsymbol{x}^{(0)}$，令

$$\boldsymbol{y}^{(0)}=\boldsymbol{x}^{(0)}/\max(\boldsymbol{x}^{(0)})$$

并定义

$$\boldsymbol{x}^{(1)}=\boldsymbol{A}\,\boldsymbol{y}^{(0)}$$

一般地，有

$$\left.\begin{aligned}\boldsymbol{y}^{(k)}&=\boldsymbol{x}^{(k)}/\max(\boldsymbol{x}^{(k)})\\ \boldsymbol{x}^{(k+1)}&=\boldsymbol{A}\,\boldsymbol{y}^{(k)}\end{aligned}\quad (k=0,1,2,\cdots)\right\} \tag{5.22}$$

当 k 足够大，并且 $\|\boldsymbol{y}^{(k+1)}-\boldsymbol{y}^{(k)}\|_1\to 0$ 时，若 $y_m^{(k)}\approx 1$，则特征值 $\lambda\approx x_m^{(k+1)}$，对应的特征向量是 $\boldsymbol{v}\approx\boldsymbol{y}^{(k)}$。称此方法为规范化的乘幂法或改进的乘幂法。

例 5.3　用规范化的乘幂法公式计算矩阵 $\boldsymbol{A}=\begin{bmatrix}0&1\\1&1\end{bmatrix}$ 的模最大的特征值及对应的特征向量。

解　取初始向量为 $\boldsymbol{x}^{(0)}=[1\quad 1]^{\mathrm{T}}$，用递推公式(5.22)计算，并进行分析，结果列入表5.2中。

表　5.2

k	$(x^{(k)})^{\mathrm{T}}$		$(y^{(k)})^{\mathrm{T}}$		k	$(x^{(k)})^{\mathrm{T}}$		$(y^{(k)})^{\mathrm{T}}$	
1	1.000 0	2.000 0	0.500 0	1.000 0	6	1.000 0	1.615 4	0.619 0	1.000 0
2	1.000 0	1.500 0	0.666 7	1.000 0	7	1.000 0	1.619 0	0.617 6	1.000 0
3	1.000 0	1.666 7	0.600 0	1.000 0	8	1.000 0	1.617 6	0.618 2	1.000 0
4	1.000 0	1.600 0	0.625 0	1.000 0	9	1.000 0	1.618 2	0.618 0	1.000 0
5	1.000 0	1.625 0	0.615 4	1.000 0	10	1.000 0	1.618 0	0.618 1	1.000 0

由表可以得出 $\lambda \approx 1.6180$，对应的特征向量是 $\boldsymbol{v} \approx [0.6181 \quad 1.0000]^{\mathrm{T}}$。

程序(5.1) 用规范化的乘幂法计算矩阵特征值及特征向量的 MATLAB 程序。

程序任务：用规范化的乘幂法计算矩阵模最大的特征值及对应的特征向量。

```
function [M V] = MatPow(A,ep,Nm)
% 输入:A—— 矩阵(N×N)
%      ep—— 计算精度
%      Nm—— 最大迭代次数
% 输出:M—— 模最大的特征值
%      V—— 与 M 对应的特征向量
n = length(A);        % 提取方阵 A 的阶
V = ones(n,1);        % 特征向量的迭代初值
k = 0;M0 = 0;         % 迭代次数控制参数及中间参量
while k < Nm
    Y = A * V;
    [Ym, i] = max(abs(Y));     % 提取绝对值最大元素的行号 i
    Ym = Y(i); V = Y/Ym;       % 特征向量模最大的元素归一化
    if abs(Ym - M0) < ep break; end
    M0 = Ym;
    k = k+1;
end
M = Ym
```

用程序(5.1)计算矩阵

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

的模最大的特征值及对应的特征向量。

在 MATLAB 命令窗口输入：

```
>> A=[2,-1,0; -1,2,-1; 0,-1,2];          % 输入矩阵
>> [M V]=MatPow(A,1e-5,50)               % 调用程序(5.1)计算
```

输出结果：

```
M =3.4142
V =[-0.7071   1.0000   -0.7071]'
```

即矩阵模最大的特征值是 3.4142，对应的特征向量为 $[-0.7071 \quad 1.0000 \quad -0.7071]'$。

5.3 反 幂 法

反幂法用于计算矩阵模最小的特征值和对应的特征向量。

若 $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$，则 $\boldsymbol{A}^{-1}\boldsymbol{v} = \boldsymbol{v}/\lambda$。因此，$\boldsymbol{A}^{-1}$ 模最大特征值的倒数正好是 $\boldsymbol{A}$ 的模最小特征值，并且对应的特征向量相同。

定义(5.5)　用乘幂法计算A^{-1}模最大的特征值$1/\lambda_1$，此时递推公式是

$$x^{(k+1)} = A^{-1} x^{(k)} \quad (k=0,1,2,\cdots) \tag{5.23}$$

则λ_1就是A的模最小的特征值，这种方法称为反幂法。

由于计算逆矩阵比较麻烦，故将递推公式改写为

$$A x^{(k+1)} = x^{(k)} \quad (k=0,1,2,\cdots) \tag{5.24}$$

通过求解线性方程组式(5.24)，从$x^{(k)}$计算出$x^{(k+1)}$，建立序列$\{x^{(k)}\}$。通常采用三角分解法求解该方程组，即先把A分解成$A=LU$，再解方程$LU x^{(k+1)} = x^{(k)}$。之后，分析序列$\{x^{(k)}\}$，得到A^{-1}的主特征值$1/\lambda_1$和对应特征向量，再得到矩阵A的模最小特征值λ_1和对应的同一特征向量。

例5.4　用反幂法计算矩阵$A=\begin{bmatrix}0 & 1\\1 & 1\end{bmatrix}$的模最小的特征值及对应的特征向量。

解　取初始向量为$x^{(0)} = [1 \quad 1]^{T}$，用递推公式(5.24)计算，并进行分析，结果列入表5.3中。

表　5.3

k	$(x^{(k)})^{T}$		$x_2^{(k)}/x_2^{(k-1)}$	k	$(x^{(k)})^{T}$		$x_2^{(k)}/x_2^{(k-1)}$
1	0	1	1	7	−8	5	−1.666 7
2	1	0	0	8	13	−8	−1.600 0
3	−1	1	1/0	9	−21	13	−1.625 0
4	2	−1	−1	10	34	−21	−1.615 4
5	−3	2	−2	11	−55	34	−1.619 0
6	5	−3	−1.5	12	89	−55	−1.617 6

由表5.3可以得出矩阵A^{-1}的主特征值是$-1.617\,6$，矩阵A的模最小的特征值是$\lambda \approx 1/(-1.617\,6) = -0.618\,2$，对应的特征向量是$v \approx [89 \quad -55]^{T}$。

程序(5.2)　用反幂法计算矩阵特征值及特征向量的MATLAB程序。

程序任务：用反幂法计算矩阵模最小的特征值及对应的特征向量。

```
function [m v] = MatInv(A,ep,Nm)
% 输入:A—— 矩阵 N×N
%      ep—— 计算精度
%      Nm—— 最大迭代次数
% 输出:m—— 模最小的特征值
%      v—— 与 m 对应的特征向量
n = length(A);      % 提取方阵 A 的阶
v = ones(n,1);      % 特征向量的迭代初值
k = 0;  m0 = 0;     % 迭代次数控制参数及中间参量
while k < Nm
    y = coluGauss(A,v);     % 用列主元素高斯消去法解线性方程(程序(4.2))
    [ym, i] = max(abs(y));
    ym = y(i);v = y/ym;     % 特征向量模最大的元素归一化
```

```
    if abs(ym - m0) < ep break; end
    m0 = y(i);
    k = k + 1;
end
m = 1/ym;
```

用程序(5.2) 计算矩阵

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

的模最小的特征值及对应的特征向量。

在 MATLAB 命令窗口输入：

```
>> A =[2, -1,0; -1,2, -1;0, -1,2];        % 输入矩阵
>> [m v] =MatInv(A,1e-5,50)               % 调用程序(5.2) 计算
```

输出结果：

```
m =0.5858
v =[0.7071   1.0000   0.7071]'
```

即矩阵模最小的特征值是 0.585 8,对应的特征向量为[0.707 1　1.000 0　0.707 1]′。

5.4　雅可比方法

设 $\boldsymbol{A}$ 为 n 阶实对称矩阵,则存在正交矩阵 $\boldsymbol{R}$,可使

$$\boldsymbol{R}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{R} = \boldsymbol{R}^{-1}\boldsymbol{A}\boldsymbol{R} = \boldsymbol{D} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$$

根据定理(5.4) 可知,对角矩阵 $\boldsymbol{D}$ 的对角元素就是矩阵 $\boldsymbol{A}$ 的全部特征值,而矩阵 $\boldsymbol{R}$ 的列向量就是矩阵 $\boldsymbol{A}$ 的全部特征向量。

雅可比方法是计算实对称矩阵全部特征值和特征向量的方法。它用代表一系列平面旋转变换的正交矩阵的相似变换把实对称矩阵对角化,从而得到实对称矩阵的特征值和特征向量。这里只介绍雅可比方法的计算过程。

(1) 令 $\boldsymbol{A}_0 = \boldsymbol{A} = [a_{ij}^{(0)}] = [a_{ij}]$,选择矩阵 $\boldsymbol{A}_0$ 的非对角元素中绝对值最大的元素(称为非对角主元素),如,$a_{i_0j_0}^{(0)}$。根据下面的公式(5.26) 和公式(5.27)(取 $k=0$) 计算旋转角 θ_0 并构造平面旋转矩阵 $\boldsymbol{R}_0(i_0, j_0)$,然后计算 $\boldsymbol{A}_1 = \boldsymbol{R}_0^{\mathrm{T}}\boldsymbol{A}_0\boldsymbol{R}_0$。

(2) 得到 $\boldsymbol{A}_k = [a_{ij}^{(k)}]$ 后,再选取其非对角主元素 $a_{i_kj_k}^{(k)}$,即

$$|a_{i_kj_k}^{(k)}| = \max_{i\neq j}\{|a_{ij}^{(k)}|\} \tag{5.25}$$

根据下面的公式(5.26) 和公式(5.27) 计算旋转角 θ_k 并构造平面旋转矩阵 $\boldsymbol{R}_k(i_k, j_k)$：

$$\left.\begin{aligned} &\tan(2\theta_k) = \frac{2a_{i_kj_k}^{(k)}}{a_{i_ki_k}^{(k)} - a_{j_kj_k}^{(k)}} \quad \left(|\theta_k| \leqslant \frac{\pi}{4}\right) \quad (a_{i_ki_k}^{(k)} - a_{j_kj_k}^{(k)} \neq 0) \\ &\theta_k = \begin{cases} +\dfrac{\pi}{4} & (a_{i_kj_k}^{(k)} > 0) \\ -\dfrac{\pi}{4} & (a_{i_kj_k}^{(k)} < 0) \end{cases} \quad (a_{i_ki_k}^{(k)} - a_{j_kj_k}^{(k)} = 0) \end{aligned}\right\} \tag{5.26}$$

$$
\boldsymbol{R}_k(i_k,j_k)=\begin{array}{c}\begin{matrix}&&&i_k\text{ 列}&&&j_k\text{ 列}&&&\end{matrix}\\\begin{bmatrix}1 &&&&&&&&& \\ &\ddots&&&&&&&& \\ &&1&&&&&&& \\ &&&\cos\theta_k&&&\sin\theta_k&&& \\ &&&&1&&&&& \\ &&&&&\ddots&&&& \\ &&&&&&1&&& \\ &&&-\sin\theta_k&&&\cos\theta_k&&& \\ &&&&&&&1&& \\ &&&&&&&&\ddots& \\ &&&&&&&&&1\end{bmatrix}\end{array}\begin{matrix} \\ \\ \\ i_k\text{ 行} \\ \\ \\ \\ j_k\text{ 行} \\ \\ \\ \\ \end{matrix} \tag{5.27}
$$

并且 $\boldsymbol{R}_k(i_k,j_k)$ 中其他元素均等于零。然后计算 $\boldsymbol{A}_{k+1}=\boldsymbol{R}_k^{\mathrm{T}}\boldsymbol{A}_k\boldsymbol{R}_k$，其中 $a_{i_kj_k}^{(k+1)}=a_{j_ki_k}^{(k+1)}=0$。另外，$\boldsymbol{A}_{k+1}$ 和 $\boldsymbol{A}_k$ 中只有 i_k 和 j_k 两行(列) 的元素不同(即 $a_{ij}^{(k+1)}=a_{ji}^{(k+1)}=a_{ij}^{(k)}\ (i,j\neq i_k,j_k)$)，它们的关系是

$$
\left.\begin{aligned}
a_{i_ki_k}^{(k+1)}&=a_{i_ki_k}^{(k)}\cos^2\theta_k+2a_{i_kj_k}^{(k)}\sin\theta_k\cos\theta_k+a_{j_kj_k}^{(k)}\sin^2\theta_k\\
a_{j_kj_k}^{(k+1)}&=a_{i_ki_k}^{(k)}\sin^2\theta_k-2a_{i_kj_k}^{(k)}\sin\theta_k\cos\theta_k+a_{j_kj_k}^{(k)}\cos^2\theta_k\\
a_{i_kj}^{(k+1)}&=a_{ji_k}^{(k+1)}=a_{ji_k}^{(k)}\cos\theta_k+a_{jj_k}^{(k)}\sin\theta_k\quad(j\neq i_k,j_k)\\
a_{ij_k}^{(k+1)}&=a_{j_ki}^{(k+1)}=-a_{ii_k}^{(k)}\sin\theta_k+a_{ij_k}^{(k)}\cos\theta_k\quad(i\neq i_k,j_k)
\end{aligned}\right\} \tag{5.28}
$$

若设

$$
y_k=|a_{i_ki_k}^{(k)}-a_{j_kj_k}^{(k)}|,\quad x_k=2\mathrm{sign}(a_{i_ki_k}^{(k)}-a_{j_kj_k}^{(k)})a_{i_kj_k}^{(k)} \tag{5.29}
$$

则式(5.26) 的第一式表示为 $\tan(2\theta_k)=x_k/y_k$，同时

$$
\cos\theta_k=\left[\frac{1}{2}\left(1+\frac{y_k}{\sqrt{x_k^2+y_k^2}}\right)\right]^{1/2},\quad \sin\theta_k=\frac{x_k}{2\cos\theta_k\sqrt{x_k^2+y_k^2}} \tag{5.30}
$$

(3) 取 $k=1,2,\cdots$，重复第(2) 步的计算，直到将矩阵 $\boldsymbol{A}$ 的非对角元素全部变换到充分小为止。

由雅可比法的收敛性可知，当 k 足够大时，有

$$
\boldsymbol{A}_{k+1}=\boldsymbol{R}_k^{\mathrm{T}}\boldsymbol{R}_{k-1}^{\mathrm{T}}\cdots\boldsymbol{R}_0^{\mathrm{T}}\boldsymbol{A}\boldsymbol{R}_0\boldsymbol{R}_1\cdots\boldsymbol{R}_k\approx\boldsymbol{D} \tag{5.31}
$$

若记

$$
\boldsymbol{P}_k=\boldsymbol{R}_0\boldsymbol{R}_1\cdots\boldsymbol{R}_k \tag{5.32}
$$

则式(5.31) 表示为 $\boldsymbol{P}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{P}_k\approx\boldsymbol{D}$，并且

$$
\boldsymbol{A}\boldsymbol{P}_k\approx\boldsymbol{P}_k\boldsymbol{D}\approx\boldsymbol{P}_k\mathrm{diag}(\lambda_1,\quad\lambda_2,\quad\cdots,\quad\lambda_n) \tag{5.33}
$$

因此，$\boldsymbol{P}_k$ 的各列向量就是矩阵 $\boldsymbol{A}$ 特征向量的近似值。当 $k\to\infty$ 时，$\boldsymbol{P}_k$ 的各列向量就是矩阵 $\boldsymbol{A}$ 的特征向量。

例 5.5　用雅可比方法计算矩阵 $\boldsymbol{A}=\begin{bmatrix}2&-1&0\\-1&2&-1\\0&-1&2\end{bmatrix}$ 的特征值和特征向量。

解　(1) 令 $\boldsymbol{A}_0=\boldsymbol{A}$。选 $\boldsymbol{A}_0=[a_{ij}^{(0)}]$ 的非对角主元素，$a_{12}^{(0)}=a_{21}^{(0)}=-1$，即 $i_0=1$ 及 $j_0=2$；

而 $a_{11}^{(0)}=a_{22}^{(0)}=2$，所以取 $\theta_0=-\pi/4$，并且

$$\boldsymbol{R}_0(1,2)=\begin{bmatrix}0.7071 & -0.7071 & 0\\ 0.7071 & 0.7071 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$\boldsymbol{A}_1=\boldsymbol{R}_0^{\mathrm{T}}\boldsymbol{A}_0\boldsymbol{R}_0=\begin{bmatrix}1 & 0 & -0.7071\\ 0 & 3 & -0.7071\\ -0.7071 & -0.7071 & 2\end{bmatrix}$$

(2) 选 $\boldsymbol{A}_1=[a_{ij}^{(1)}]$ 的非对角主元素，$a_{31}^{(1)}=a_{13}^{(1)}=-0.7071$，即 $i_1=3$ 及 $j_1=1$；而 $a_{33}^{(1)}=2$ 及 $a_{11}^{(1)}=1$，所以

$$\tan(2\theta_1)=\frac{2(-0.7071)}{2-1}=-1.4142$$

取 $\theta_1=-27.3678°$ 并且 $\sin\theta_1=-0.4597$ 及 $\cos\theta_1=0.8881$，因此

$$\boldsymbol{R}_1(1,3)=\begin{bmatrix}0.8881 & 0 & -0.4597\\ 0 & 1 & 0\\ 0.4597 & 0 & 0.8881\end{bmatrix}$$

$$\boldsymbol{A}_2=\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{A}_1\boldsymbol{R}_1=\begin{bmatrix}0.6340 & -0.3251 & 0\\ -0.3251 & 3 & -0.6280\\ 0 & -0.6280 & 2.3660\end{bmatrix}$$

(3) 选 $\boldsymbol{A}_2=[a_{ij}^{(2)}]$ 的非对角主元素，$a_{32}^{(2)}=a_{23}^{(2)}=-0.6280$，即 $i_2=3$ 及 $j_2=2$；而 $a_{33}^{(2)}=2.3660$ 及 $a_{22}^{(2)}=3$，以及

$$\tan(2\theta_2)=\frac{2(-0.6280)}{2.3660-3}=1.9811$$

取 $\theta_2=31.608°$ 并且 $\sin\theta_2=0.5241$ 及 $\cos\theta_2=0.8517$，因此

$$\boldsymbol{R}_2(2,3)=\begin{bmatrix}1 & 0 & 0\\ 0 & 0.8517 & 0.5241\\ 0 & -0.5241 & 0.8517\end{bmatrix}$$

$$\boldsymbol{A}_3=\boldsymbol{R}_2^{\mathrm{T}}\boldsymbol{A}_2\boldsymbol{R}_2=\begin{bmatrix}0.6340 & -0.2768 & -0.1704\\ -0.2768 & 3.3864 & -0.0000\\ -0.1704 & -0.0000 & 1.9796\end{bmatrix}$$

(4) 反复计算。当 $k=6$ 时，有

$$\boldsymbol{A}_7=\begin{bmatrix}0.5858 & 0.0000 & 0.0000\\ 0.0000 & 3.4142 & 0.0000\\ 0.0000 & 0.0000 & 2.0000\end{bmatrix}$$

$$\boldsymbol{P}_7=\begin{bmatrix}0.5000 & -0.5000 & -0.7071\\ 0.7071 & 0.7071 & 0.0000\\ 0.5000 & -0.5000 & 0.7071\end{bmatrix}$$

即 $\boldsymbol{A}$ 的三个特征值近似是

$$\lambda_1\approx 0.5858,\quad \lambda_2\approx 3.4142,\quad \lambda_3\approx 2.0000$$

对应的特征向量分别是

$$v_1 = [0.500\,0 \quad 0.707\,1 \quad 0.500\,0]^T$$
$$v_2 = [-0.500\,0 \quad 0.707\,1 \quad -0.500\,0]^T$$
$$v_3 = [-0.707\,1 \quad 0.000\,0 \quad 0.707\,1]^T$$

程序(5.3)　求解矩阵特征值和特征向量的雅可比方法的 MATLAB 程序。

程序任务：用雅可比方法求解实对称矩阵的特征值和特征向量。

```
function [R D] = MatJaco (A,ep)
% 输入:A—— 实对称矩阵(N×N)
%      ep—— 计算精度
% 输出:R—— 特征向量组成的矩阵(N×N)
%      D—— 特征值组成的对角矩阵(N×N)
n = length(A);          % 提取方阵 A 的阶
P = eye(n);             % 初始化矩阵 P
while 1
    % 提取非对角主元素的行 i 和列 j(行号优先)
    Am = 0;
    for l = 1:(n-1)
        for m = (l+1):n
            if abs(A(l,m)) > Am
                Am = abs(A(l,m));
                i = l; j = m;
            end
        end
    end
    if Am < ep break; end                      % 非对角主元素的绝对值小于 ep 时,计算终断
    % 计算转角的三角函数值
    Co = (A(i,i) - A(j,j))/(2 * A(i,j));      % 转角两倍的余切值
    if abs(Co) < 1e-10
        t = 1;                                 % 当 Co 过小时取转角为 45°
    else
        t = sign(Co)/(abs(Co) + sqrt(Co^2 + 1))% 计算转角的正切值
    end
    c = 1/sqrt(t^2 + 1); s = c * t;            % 转角的余弦值和正弦值
    % 建立转动矩阵 Rk
    R = eye(n);
    R(i,i) = c; R(i,j) = -s; R(j,i) = s; R(j,j) = c;
    A = R' * A * R;    % 实施相似变换
    P = P * R;         % 计算总的相似变换矩阵
end
R = P;
D = diag(diag(A));
```

用程序(5.3)计算矩阵

$$A=\begin{bmatrix}8 & -1 & 3 & -1\\ -1 & 6 & 2 & 0\\ 3 & 2 & 9 & 1\\ -1 & 0 & 1 & 7\end{bmatrix}$$

的特征值与特征向量。

在 MATLAB 命令窗口输入：

```
>> A=[8,-1,3,-1;-1,6,2,0;3,2,9,1;-1,0,1,7];    % 输入方阵 A
>> [R D]=MatJaco(A,1e-4)                          % 调用程序(5.3)计算
```

输出结果：

```
R =
     0.5288   -0.5730   0.5823    0.2301
     0.5920    0.4723   0.1758   -0.6290
    -0.5360    0.2820   0.7925   -0.0712
     0.2874    0.6075   0.0447    0.7392
D =
    3.2957         0         0         0
         0    8.4077         0         0
         0         0   11.7043         0
         0         0         0    6.5923
```

即 $\boldsymbol{A}$ 的特征值近似是

$$\lambda_1\approx 3.295\,7,\quad \lambda_2\approx 8.407\,7,\quad \lambda_3\approx 11.704\,3,\quad \lambda_4\approx 6.592\,3$$

对应的特征向量分别是

$$\boldsymbol{v}_1=[0.528\,8\quad 0.592\,1\quad -0.536\,0\quad 0.287\,4]^{\mathrm{T}}$$
$$\boldsymbol{v}_2=[-0.573\,0\quad 0.472\,3\quad 0.282\,0\quad 0.607\,5]^{\mathrm{T}}$$
$$\boldsymbol{v}_3=[0.582\,3\quad 0.175\,8\quad 0.792\,5\quad 0.044\,7]^{\mathrm{T}}$$
$$\boldsymbol{v}_4=[0.230\,1\quad -0.629\,0\quad -0.071\,2\quad 0.739\,2]^{\mathrm{T}}$$

5.5 物理学中的应用举例——简单剪切变形的主应变分析

图 5.1 所示是一个长方体的简单剪切变形。变形前长方体的各个棱与坐标轴 $X_i(i=1,2,3)$ 平行，竖向的四个棱在变形后与坐标轴 X_2 的夹角均为 β，水平棱在变形后方向不变。变形前长方体中坐标为 (X_1,X_2,X_3) 的点，变形后坐标是

$$x_1=X_1+X_2\tan\beta,\quad x_2=X_2,\quad x_3=X_3 \tag{5.34}$$

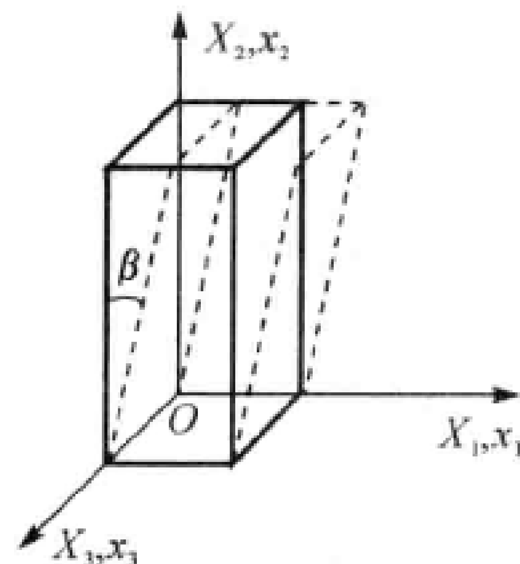


图 5.1

该点的右柯西-格林变形张量和拉格朗日应变张量分别是

$$\boldsymbol{C}=\begin{bmatrix}1 & \tan\beta & 0\\ \tan\beta & 1+\tan^2\beta & 0\\ 0 & 0 & 1\end{bmatrix},\quad \boldsymbol{E}=\frac{1}{2}(\boldsymbol{C}-\boldsymbol{I})=\frac{1}{2}\begin{bmatrix}0 & \tan\beta & 0\\ \tan\beta & \tan^2\beta & 0\\ 0 & 0 & 0\end{bmatrix} \tag{5.35}$$

设拉格朗日应变张量 $\boldsymbol{E}$ 的特征值(主值)为 $E_i(i=1,2,3)$,与之对应的单位特征向量(主方向)为 $\boldsymbol{e}_i(i=1,2,3)$,则变形前沿方向 $\boldsymbol{e}_i$、长度为 $\mathrm{d}L_i$ 的线元在瞬时变形中方向不变、长度变为 $\mathrm{d}l_i$,其伸长比为

$$\lambda_i=\frac{\mathrm{d}l_i}{\mathrm{d}L_i}=\sqrt{1+2E_i}\quad(i=1,2,3)\tag{5.36}$$

主方向 $\boldsymbol{e}_i$ 与 X_1 轴之间的夹角 θ_i 满足

$$\cos\theta_i=\boldsymbol{e}_i\cdot\boldsymbol{n}_1\quad(i=1,2,3)\tag{5.37}$$

式中,$\boldsymbol{n}_1$ 是 X_1 轴方向的单位矢量。

可以应用雅可比方法,计算 β 取不同值时应变张量 $\boldsymbol{E}$ 的特征值(主值)与特征向量(主方向),据此分析简单剪切变形过程中应变张量主值与主方向随 β 的变化情况。下面是数值计算与曲线绘制程序。

程序(5.4)　简单剪切变形过程中拉格朗日应变张量主值与主方向分析的 MATLAB 程序。

```
% 简单剪切变形过程中拉格朗日应变张量主值与主方向分析程序
nx = [1,0,0];            %x1 轴方向的单位矢量
for i = 1:80;            % 角 β 的取值范围(单位为度)
    bt = (pi/180) * i;   % 角 β 的值化为弧度
    beta(i) = i;         % 形成作图时的横轴坐标向量
    E = (1/2) * [0,tan(bt),0;tan(bt),tan(bt)^2,0;0,0,0];    % 建立应变张量 E
    [R D] = MatJaco(E,1e-5);              % 用雅可比方法计算 E 的特征值与特征向量(程序(5.3))
    u1 = R(:,1); E1 = D(1,1);             % 提取特征向量ui 与特征值 Ei (i = 1,2,3)
    u2 = R(:,2); E2 = D(2,2);
    u3 = R(:,3); E3 = D(3,3);
    e1 = u1/sqrt(sum(u1.^2));             % 特征向量归一化
    e2 = u2/sqrt(sum(u2.^2));
    e3 = u3/sqrt(sum(u3.^2));
    ae1(i) = asin(dot(cross(nx,e1'),[0,0,1])) * 180/pi;    % 特征向量与 x1 轴的夹角 θi(度)
    ae2(i) = asin(dot(cross(nx,e2'),[0,0,1])) * 180/pi;
    ae3(i) = acos(dot(nx,e3')) * 180/pi;
    lamda1(i) = sqrt(1 + 2 * E1);         % 计算伸长比 λi (i = 1,2,3)
    lamdaIJ(i) = sqrt(1+2 * E2);
    lamda3(i) = sqrt(1 + 2 * E3);
end
figure;              % 绘制 θi -β 曲线
plot(beta,ae1,'-k',beta,ae2,'--k',beta,ae3,':k');
axis([0 max(beta) min([ae1(:);ae2(:);ae3(:)]) max([ae1(:);ae2(:);ae3(:)]) + 10]);
xlabel('\beta'); ylabel('\theta_i');
gtext('\theta_1'); gtext('\theta_2'); gtext('\theta_3');
figure;              % 绘制 λi -β 曲线
plot(beta,lamda1,'-k',beta,lamda2,'--k',beta,lamda3,':k');
axis([0 max(beta) min([lamda1(:);lamda2(:);lamda3(:)])
max([lamda1(:);lamda2(:);lamda3(:)])]);
```

```
xlabel('\beta'); ylabel('\lambda_i');
gtext('\lambda_1'); gtext('\lambda_2'); gtext('\lambda_3');
```

运行程序，绘制的曲线如图 5.2 和图 5.3 所示。分析这些曲线，可以得到以下结论：

(1) 主方向 1 处在 X_1OX_2 平面内，与 X_1 轴夹角 θ_1 为负，方向偏向 X_2 轴负方向。随着 β 的增大 $|\theta_1|$ 减小，说明主方向 1 逐渐向 X_1 方向靠近。沿着主方向 1 的伸长比 $\lambda_1 < 1$，并且随着 β 的增大而减小，可见沿此方向的线元被压缩得越来越厉害。

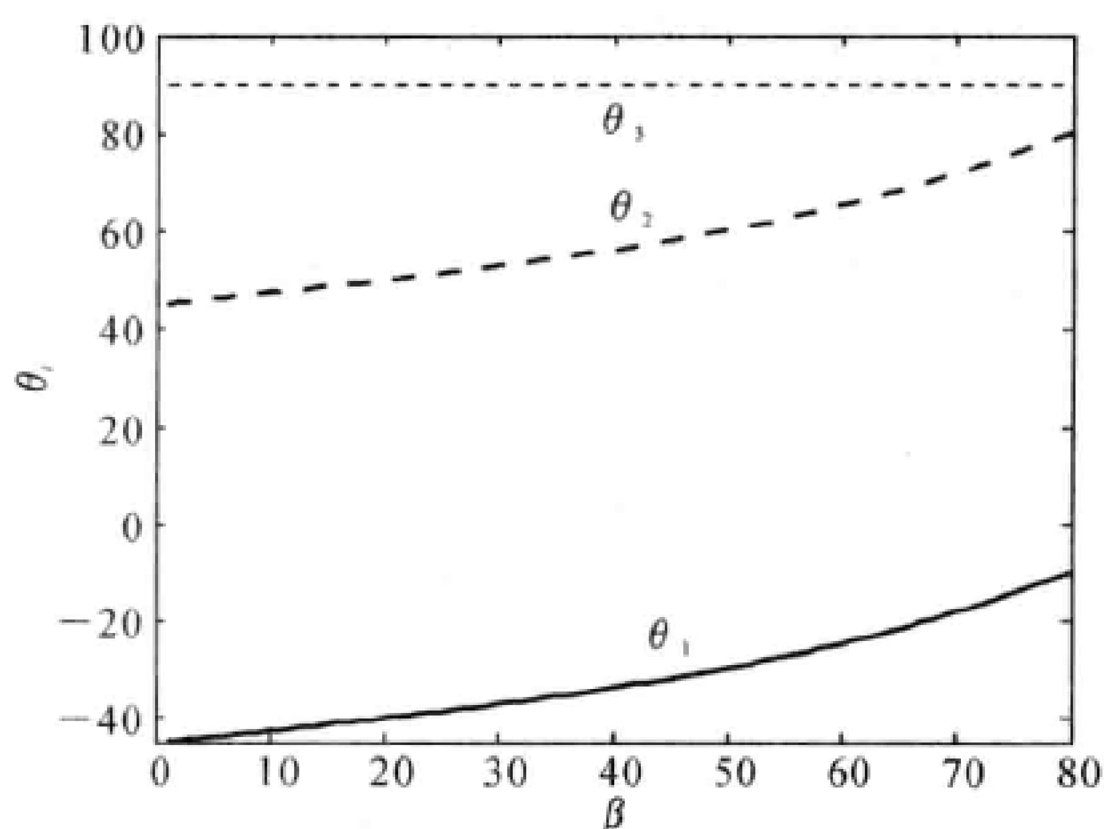


图 5.2

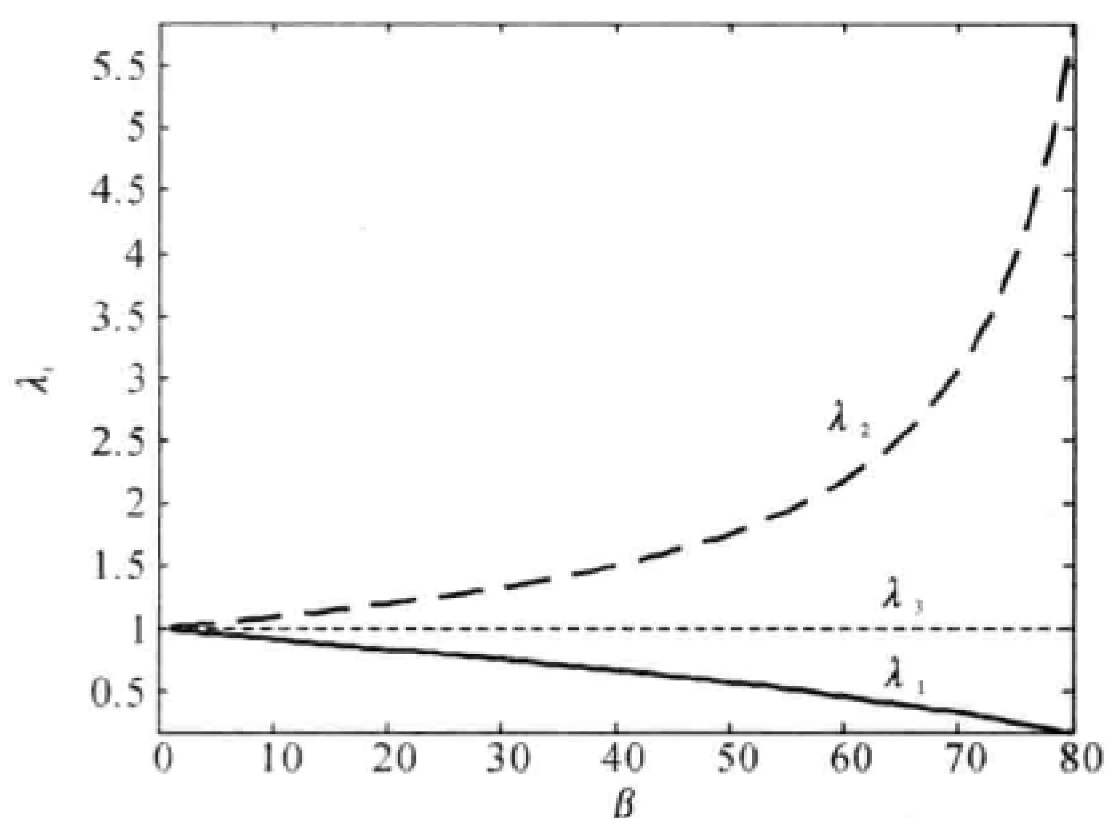


图 5.3

(2) 主方向 2 处在 X_1OX_2 平面内，与 X_1 轴夹角 θ_2 为正，方向偏向 X_2 轴正方向。随着 β 的增大 θ_2 增大，说明主方向 2 逐渐偏离 X_1 方向。沿着主方向 2 的伸长比 $\lambda_2 > 1$，并且随着 β 的增大而增大，可见沿此方向的线元被拉伸得越来越严重。

(3) 主方向 3 垂直于 X_1OX_2 平面，与 X_1 轴夹角 θ_3 恒等于 90°。沿着主方向 3 的伸长比 $\lambda_3 = 1$，所以沿此方向的线元长度不变。

(4) θ_1 与 θ_2 曲线走势相同，竖向间距不变。这是因为主方向 1 与主方向 2 相互垂直，夹角始终为 90°。

习　　题

1. 取$x^{(0)}=[0\quad 0\quad 1]^T$,用乘幂法计算矩阵$A=\begin{bmatrix}2 & 3 & 2\\10 & 3 & 4\\3 & 6 & 1\end{bmatrix}$模最大的特征值和对应的特征向量。

2. 取$x^{(0)}=[1\quad 1\quad 1]^T$,用反幂法计算矩阵$A=\begin{bmatrix}2 & 8 & 9\\8 & 3 & 4\\9 & 4 & 7\end{bmatrix}$模最小的特征值和对应的特征向量。

3. 取$x^{(0)}=[1\quad 0]^T$,分别用乘幂法和反幂法计算矩阵$A=\begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}$的特征值和对应的特征向量。

4. 用雅可比方法编程计算矩阵$A=\begin{bmatrix}4 & 3 & 2 & 1\\3 & 4 & 3 & 2\\2 & 3 & 4 & 3\\1 & 2 & 3 & 4\end{bmatrix}$的特征值与特征向量，要求精度为1e－5。

5. 设无阻尼质量弹簧系统如图 5.4 所示。描述系统静态平衡态位置的数学模型如下：

$$\begin{bmatrix}k_1+k_2 & -k_2 & 0\\-k_2 & k_2+k_3 & -k_3\\0 & -k_3 & k_3\end{bmatrix}\begin{bmatrix}x_1(t)\\x_2(t)\\x_3(t)\end{bmatrix}+\begin{bmatrix}m_1 & 0 & 0\\0 & m_2 & 0\\0 & 0 & m_3\end{bmatrix}\begin{bmatrix}x''_1(t)\\x''_2(t)\\x''_3(t)\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

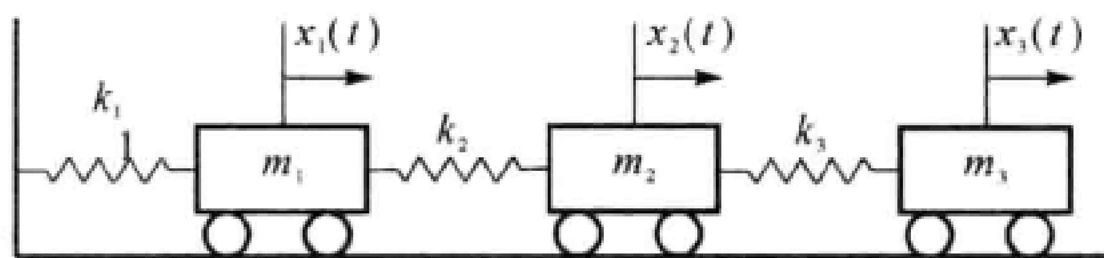


图　5.4

(1) 使用替换 $x_j(t)=v_j\sin(\omega t+\theta)$,其中 $j=1,2,3,\theta$ 是常量,证明数学模型的解可以表示为

$$\begin{bmatrix}\dfrac{k_1+k_2}{m_1} & \dfrac{-k_2}{m_1} & 0\\ \dfrac{-k_2}{m_2} & \dfrac{k_2+k_3}{m_2} & \dfrac{-k_3}{m_2}\\ 0 & \dfrac{-k_3}{m_3} & \dfrac{k_3}{m_3}\end{bmatrix}\begin{bmatrix}v_1\\v_2\\v_3\end{bmatrix}=\omega^2\begin{bmatrix}v_1\\v_2\\v_3\end{bmatrix}$$

(2) 令 $\lambda=\omega^2$，则 (1) 中方程的三个解是特征值 λ_j 和相应的特征向量 $v_j=[v_1^{(j)}\quad v_2^{(j)}\quad v_3^{(j)}]^T$,这里 $j=1,2,3$。证明它们可用来形成三个基本解：

$$\boldsymbol{x}_j(t)=\begin{bmatrix} v_1^{(j)}\sin(\omega_j t+\theta) \\ v_2^{(j)}\sin(\omega_j t+\theta) \\ v_3^{(j)}\sin(\omega_j t+\theta) \end{bmatrix}=\sin(\omega_j t+\theta)\begin{bmatrix} v_1^{(j)} \\ v_2^{(j)} \\ v_3^{(j)} \end{bmatrix}$$

其中 $\omega_j=\sqrt{\lambda_j}\,(j=1,2,3)$。这三个基本解称为振荡的三个基本模型。

(3) 根据下列参数,编程求解特征值、特征向量及振荡的三个基本模型。

1) $k_1=3, k_2=2, k_3=1, m_1=1, m_2=1, m_3=1$;

2) $k_1=0.5, k_2=0.25, k_3=0.25, m_1=4, m_2=4, m_3=4$;

3) $k_1=0.2, k_2=0.4, k_3=0.3, m_1=2.5, m_2=2.5, m_3=2.5$。

6. 设入射光波的偏振态是 $\boldsymbol{E}_i=[E_{ix}\quad E_{iy}]^{\mathrm{T}}$,通过一光学器件后,其偏振态变为 $\boldsymbol{E}_t=[E_{tx}\quad E_{ty}]^{\mathrm{T}}$,可以认为光学器件对入射光波的偏振态起到了一种变换作用。数学上把这种变换用一个变换矩阵表示,即

$$\boldsymbol{E}_t=\begin{bmatrix} E_{tx} \\ E_{ty} \end{bmatrix}=\boldsymbol{J}\boldsymbol{E}_i=\begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix}\begin{bmatrix} E_{ix} \\ E_{iy} \end{bmatrix}$$

其中,变换矩阵 $\boldsymbol{J}=\begin{bmatrix} J_{xx} & J_{xy} \\ J_{yx} & J_{yy} \end{bmatrix}$ 称琼斯矩阵。

振幅为 E_0、振动方向与 x 轴夹角为 α 的线偏振光的偏振态是 $\boldsymbol{E}_p=E_0\,[\cos\alpha\quad \sin\alpha]^{\mathrm{T}}$。透光轴与 x 轴夹角为 θ 的起偏器的琼斯矩阵是

$$\boldsymbol{J}_\theta=\begin{bmatrix} \cos^2\theta & \sin\theta\cos\theta \\ \sin\theta\cos\theta & \sin^2\theta \end{bmatrix}$$

取 $\theta=\pi/3$ 或 $\theta=\pi/4$,试分别用乘幂法(或反幂法)和雅可比方法计算琼斯矩阵的特征值及对应的特征向量,并说明特征值与特征向量的物理意义。

7. 若物体内一点处的应力张量是 $\boldsymbol{\sigma}=\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$,其中 $\sigma_{ij}=\sigma_{ji}\,(i,j=1,2,3)$,则过该点、以单位矢量 $\boldsymbol{n}=[n_1\quad n_2\quad n_3]^{\mathrm{T}}$ 为法线的平面上的应力矢量为 $\boldsymbol{T}=\boldsymbol{\sigma}\boldsymbol{n}$。

设物体内某点的应力张量为 $\boldsymbol{\sigma}=\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 1 \end{bmatrix}$,试计算该应力张量的主应力和主方向(主应力和主方向分别是应力张量矩阵的特征值和特征向量表示的方向),并说明它们的物理意义。

8. 设某晶体的相对介电张量是 $\boldsymbol{\varepsilon}_r=\begin{bmatrix} 5.3160 & -0.1338 & -0.1639 \\ -0.1338 & 5.3706 & -0.1338 \\ -0.1639 & -0.1338 & 5.3160 \end{bmatrix}$,用雅可比方法计算晶体的介电主轴方向和主相对介电常数(即介电张量矩阵的特征向量方向和特征值)。

第6章 非线性方程根的数值求解

在科学研究和工程计算中经常会遇到非线性方程根的求解问题，无论是高次代数方程，还是超越方程，求根问题都是一个复杂问题，没有具体的计算公式。因此，对于理论研究或实际应用，方程根的数值求解方法都有比较重要的意义。本章介绍非线性方程根的几种常用数值求解方法，有区间对分法、普通迭代法、牛顿迭代法、弦截法，最后还给出了解非线性方程组的迭代法。

6.1 对　分　法

非线性方程的形式是

$$f(x)=0 \tag{6.1}$$

其中，$f(x)$ 是实变量 x 的非线性函数。满足方程 $f(x)=0$ 的数 x^* 称为方程的根，也称为函数 $y=f(x)$ 的零点。

若 $f(x)$ 是代数多项式，则称方程式(6.1)为代数方程，例如 $x^2-5x+2=0$。若 $f(x)$ 含有超越函数(如三角函数、对数函数、指数函数等)，则称方程式(6.1)为超越方程，例如 $2^x-3x+1=0$。

如果 $f(x)$ 可以分解为

$$f(x)=(x-x^*)^m g(x) \tag{6.2}$$

其中 $0<|g(x^*)|<\infty$，m 为正整数，则称 x^* 为方程的 m 重根或函数 $y=f(x)$ 的 m 重零点。当 $m=1$ 时，称 x^* 为方程的单根或函数的单重零点。

求解非线性方程根的近似值问题主要包括三个方面：一是根的存在性，即方程有没有根，有几个根；二是根的分布，就是找出根所在的区间；三是求根方法，亦即把近似根精确化的方法。关于第一个问题有下面的定理：

定理(6.1)　设函数 $f(x)$ 在区间 $[a,b]$ 上连续，并且 $f(a)\cdot f(b)<0$，则存在 $x^*\in[a,b]$，使 $f(x^*)=0$，即方程 $f(x)=0$ 在 (a,b) 内至少有一个根 x^*。

可以先用搜索法来确定方程根所在的区间，再将区间 $[a,b]$ 分成 n 等份，每个区间的长度为 $\Delta x=(b-a)/n$，区间端点的坐标是 $x_i=a+i\Delta x(i=0,1,\cdots,n)$，如果

$$f(x_i)\cdot f(x_{i+1})<0 \tag{6.3}$$

那么在区间 $[x_i,x_{i+1}]$ 必有方程的根。也可以画出函数 $y=f(x)$ 的曲线，根据曲线与 x 轴的交点位置确定函数零点的大致位置和所在区间。

定义(6.1)　对分法又称为二分法。它通过逐步对分方程根所在的区间，连续对半缩小方程根所在区间范围，来搜索方程根的近似值。

设 $f(x)$ 在区间 $[a,b]$ 上连续，并且 $f(a)\cdot f(b)<0$，则方程 $f(x)=0$ 在区间 (a,b) 内至少有一个根。为了简单起见，这里不妨讨论方程 $f(x)=0$ 在区间 (a,b) 内只有一个根的情况。

1. 求根方法

(1) 取区间$[a,b]$的中点,记作$x_0=(a+b)/2$,计算函数值$f(x_0)$。若$f(x_0)=0$,则方程的根$x^*=x_0$。若$f(x_0)\cdot f(b)<0$,记$a_1=x_0,b_1=b$;若$f(x_0)\cdot f(b)>0$,记$a_1=a,b_1=x_0$;则$x^*\in[a_1,b_1]$。

(2) 取区间$[a_1,b_1]$的中点,记作$x_1=(a_1+b_1)/2$,计算函数值$f(x_1)$。若$f(x_1)=0$,则方程的根$x^*=x_1$。若$f(x_1)\cdot f(b_1)<0$,记$a_2=x_1,b_2=b_1$;若$f(x_1)\cdot f(b_1)>0$,记$a_2=a_1$,$b_2=x_1$;则$x^*\in[a_2,b_2]$。

(3) 当$x^*\in[a_k,b_k]$时,取区间$[a_k,b_k]$的中点,记作$x_k=(a_k+b_k)/2$,计算函数值$f(x_k)$。若$f(x_k)=0$,则方程的根$x^*=x_k$。若$f(x_k)\cdot f(b_k)<0$,记$a_{k+1}=x_k,b_{k+1}=b_k$;若$f(x_k)\cdot f(b_k)>0$,记$a_{k+1}=a_k,b_{k+1}=x_k$;则$x^*\in[a_{k+1},b_{k+1}]$。

如此连续对方程根所在区间进行对分,就得到一系列不断缩小的有根区间$[a,b]$,$[a_1,b_1]$,$[a_2,b_2]$,…,$[a_k,b_k]$,…,而每个区间落在前一个区间内,并且长度只是前一个区间的一半。

2. 终止区间对分的条件

(1) 根据根近似值的误差判定。对分过程中得到的区间套为

$$x^*\in[a_k,b_k]\subset[a_{k-1},b_{k-1}]\subset\cdots\subset[a_1,b_1]\subset[a,b] \tag{6.4}$$

区间$[a_k,b_k]$的长度是

$$b_k-a_k=\frac{1}{2}(b_{k-1}-a_{k-1})=\frac{1}{2^k}(b-a) \tag{6.5}$$

若取区间$[a_k,b_k]$的中点坐标$x_k=(a_k+b_k)/2$为方程根的近似值,则误差限为

$$|x_k-x^*|\leqslant\frac{1}{2}(b_k-a_k)=\frac{1}{2^{k+1}}(b-a) \tag{6.6}$$

因此,只要满足

$$\frac{1}{2^{k+1}}(b-a)\leqslant\varepsilon \tag{6.7}$$

即可得到满足精度ε要求的方程根近似解

$$x^*\approx\frac{a_k+b_k}{2} \tag{6.8}$$

利用式(6.7),可以求出对分区间的最少次数

$$k_{\min}=\frac{\ln[(b-a)/\varepsilon]}{\ln 2}-1 \tag{6.9}$$

(2) 利用函数值判定。当根的近似值满足给定的函数值大小要求

$$|f(x_k)|<\varepsilon' \tag{6.10}$$

时,就可以取方程根的近似值为$x^*\approx x_k$。

(3) 利用区间长度判定。根据公式(6.7),当区间$[a_k,b_k]$的长度$b_k-a_k\leqslant 2\varepsilon$时,取方程根的近似值为$x^*\approx(a_k+b_k)/2$,就可满足误差要求$|x_k-x^*|\leqslant\varepsilon$。

对分法的优点是方法简单、计算程序容易编制,并且对函数$f(x)$要求不高,但该方法只能用于求解方程的实根,不能求解复根和偶数重根。对分法的求解过程如图6.1所示。

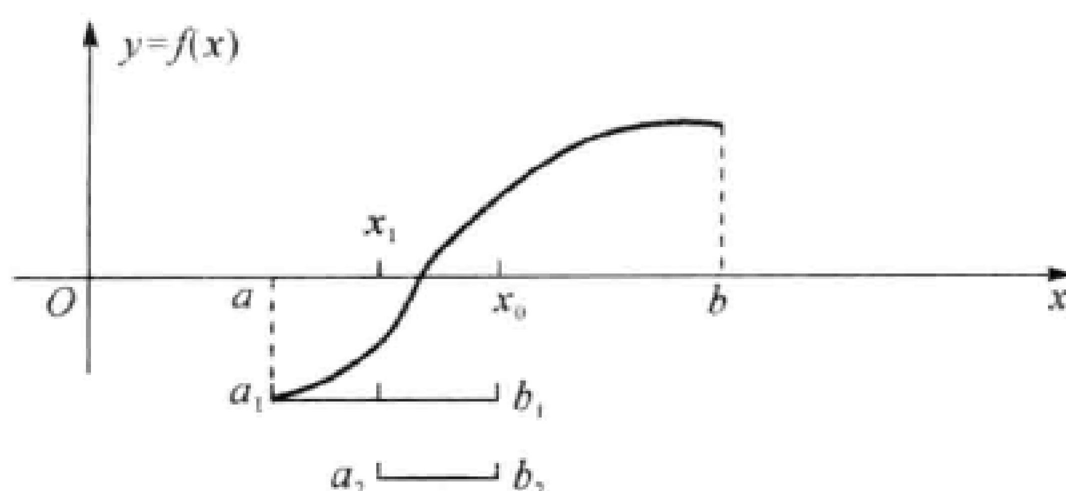


图　6.1

例 6.1　用对分法求方程 $x^3-2x-5=0$ 在区间 [2,3] 内的实根近似值，并指出其误差限。

解　函数 $f(x)=x^3-2x-5$，区间 [2,3] 的端点是 $a=2,b=3$。显然函数 $f(x)$ 在区间 [2,3] 上连续，并且 $f(a)\cdot f(b)=-1\times 16<0$，所以方程 $f(x)=0$ 在区间 [2,3] 内有根。

用对分法求根的计算结果见表 6.1。

表　6.1

k	a_k	b_k	x_k	$f(x_k)$ 的符号
0	2	3	2.5	+
1	2	2.5	2.25	+
2	2	2.25	2.125	+
3	2	2.21 5	2.062 5	−
4	2.062 5	2.215	2.093 75	−
5	2.093 75	2.215	2.109 375	+
6	2.093 75	2.109 375	2.101 562 5	

取解的近似值为 $x^*\approx x_6=2.101\ 562\ 5$，其误差限为

$$|x^*-x_6|\leqslant\frac{3-2}{2^{6+1}}=0.078\ 125$$

（方程的精确解为 $x^*=2.094\ 551\ 5$）。

程序(6.1)　用对分法计算方程根近似值的 MATLAB 程序。

程序任务：用对分法计算方程 $f(x)=0$ 的根的近似值。

```
function [x0, k] = RootDiv(equat, a, b, ep)
% 输入:equat—— 求根的函数 f(x)
%       a—— 根所在区间的下限
%       b—— 根所在区间的上限
%       ep—— 计算精度(当根所在区间的一半小于 ep 时计算结束,并以区间中点坐标作为根的近似值)
% 输出:x0—— 根的近似值
%       k—— 迭代次数(输出 k = 0 时,表示在区间[a,b] 内方程无解,同时输出 x0 = [f(a) ,f(b) ])
fa = feval(equat, a); fb = feval(equat, b);        % 计算值 f(a) 和 f(b)
if fa * fb > 0 x0 = [fa, fb]; k = 0; return; end
```

```
k = 1;
while abs(b - a)/2 > ep
  k = k + 1;
  x = (a + b)/2; fx = feval(equat, x);          % 计算区间[a, b]的中点 x 及 f(x)
  if fx * fa < 0
    b = x; fb = fx;                             % f(b)f(x) < 0 时,区间[a, b]变为[a, x]
  else
    a = x; fa = fx;                             % f(b)f(x) > 0 时,区间[a, b]变为[x, b]
  end
end
x0 = (a + b)/2;                                 % 方程根近似值的计算结果
```

用程序(6.1)计算方程 $f(x)=\sin x - x^2/4=0$ 在区间[1.6, 2]内的根。

在 MATLAB 命令窗口输入:

```
>> [x0, k] = RootDiv('funroot1', 1.5, 2, 1e-5)      % 调用程序(6.1)计算
```

输出结果:

```
x0 = 1.9338
k = 15
```

即方程根的近似值是 1.933 8,迭代次数为 15。计算过程中调用了函数 $f(x)=\sin x - x^2/4$ 定义文件 funroot1.m,其内容是

```
function y = funroot1(x)
y = sin(x) - x^2/4;
```

若在 MATLAB 命令窗口输入:

```
>> [x0, k] = RootDiv('funroot1', 2, 2.5, 1e-5)
```

则输出结果:

```
x0 =  -0.090 7   -0.964 0
k = 0
```

说明方程在区间[2, 2.5]内无根,因为 $f(a)f(b)>0$。

6.2 迭 代 法

6.2.1 方法介绍

定义(6.2) 迭代法是一种逐次逼近的方法,其基本思路是用一个固定的迭代公式反复校正根的近似值,从而形成一个近似值序列 $\{x_k\}$,该序列的极限就是方程 $f(x)=0$ 的根 x^*。

迭代法的求解过程:

(1) 对于给定的非线性方程 $f(x)=0$,先将其转换成等价方程(同解方程)

$$x=\varphi(x) \tag{6.11}$$

函数 $\varphi(x)$ 称为迭代函数;

(2) 然后构造迭代格式(亦称迭代公式)

$$x_{k+1}=\varphi(x_k)\quad (k=0,1,2,\cdots) \tag{6.12}$$

(3) 选定迭代初值(初始值)x_0 后，利用迭代公式构造迭代序列$\{x_k\}$。

如果函数 $\varphi(x)$ 连续，并且迭代序列收敛，即 $x^*=\lim\limits_{k\to\infty}x_{k+1}=\lim\limits_{k\to\infty}\varphi(x_k)=\varphi(\lim\limits_{k\to\infty}x_k)=\varphi(x^*)$，则 x^* 就是方程 $f(x)=0$ 的根。此时称迭代方法收敛，否则称为发散。

在迭代过程中，利用连续两次近似值差的绝对值$|x_{k+1}-x_k|$小于给定的控制精度来确定迭代的终止，并取 x_{k+1} 为方程根的近似值。

6.2.2 迭代法的收敛性

对于同一个函数 $f(x)$，可以选取不同的迭代函数 $\varphi(x)$，形成不同的迭代序列$\{x_k\}$。有的迭代序列收敛，有的迭代序列不收敛，即是收敛的迭代序列，也有收敛快慢的差异。迭代法的收敛性取决于迭代函数 $\varphi(x)$ 在有根区间内的性态。

定理(6.2)(收敛性定理)　设迭代函数 $\varphi(x)$ 在$[a,b]$上有连续的一阶导数，并且

(1) 当 $x\in[a,b]$时，$\varphi(x)\in[a,b]$；

(2) 对于任意 $x\in[a,b]$，存在正数 $q<1$，使$|\varphi'(x)|\leqslant q$ 成立。

则

(1) 方程 $x=\varphi(x)$ 在区间$[a,b]$上有唯一的根x^*，并且对于任意初始值 $x_0\in[a,b]$，利用迭代公式 $x_{k+1}=\varphi(x_k)$ 所形成的序列$\{x_k\}$均收敛于方程的根 x^*；

(2) $|x_k-x^*|\leqslant\dfrac{1}{1-q}|x_{k+1}-x_k|$；　(6.13)

(3) $|x_k-x^*|\leqslant\dfrac{q^k}{1-q}|x_1-x_0|$。　(6.14)

证明　先证明方程根 x^* 的存在性。构造函数 $g(x)=x-\varphi(x)$，则 $g(x)$ 在区间$[a,b]$上连续，由条件(1) 有

$$g(a)=a-\varphi(a)\leqslant 0,\quad g(b)=b-\varphi(b)\geqslant 0$$

故存在 $x^*\in[a,b]$，使 $g(x^*)=0$，即 $x^*=\varphi(x^*)$。

再证明根 x^* 的唯一性。设方程 $x=\varphi(x)$ 在$[a,b]$上有两个根x^* 和 $x^\#$，则根据微分中值定理和条件(2) 可得

$$|x^*-x^\#|=|\varphi(x^*)-\varphi(x^\#)|=|\varphi'(\xi)|\cdot|x^*-x^\#|\leqslant q|x^*-x^\#|$$

其中 ξ 在 x^* 和 $x^\#$ 之间。由于 $q<1$，上式只有在 $x^*=x^\#$ 时成立，故方程的根唯一。

最后证明迭代法的收敛性。条件(1) 说明，当 $x_0\in[a,b]$时，$x_k\in[a,b]$ $(k=1,2,\cdots)$。并且

$$x^*-x_{k+1}=\varphi(x^*)-\varphi(x_k)=\varphi'(\xi_k)(x^*-x_k)$$

其中 ξ_k 在 x^* 和 x_k 之间。进一步有

$$0\leqslant|x^*-x_k|\leqslant q|x^*-x_{k-1}|\leqslant q^2|x^*-x_{k-2}|\leqslant\cdots\leqslant q^k|x^*-x_0|\quad (k=1,2,\cdots)$$

由于 $q<1$，所以$\lim\limits_{k\to\infty}q^k|x^*-x_0|=0$，从而有$\lim\limits_{k\to\infty}x_k=x^*$。

由于

$$\begin{aligned}|x^*-x_k|&=|(x^*-x_{k+1})+(x_{k+1}-x_k)|\leqslant|x^*-x_{k+1}|+|x_{k+1}-x_k|\leqslant\\&q|x^*-x_k|+|x_{k+1}-x_k|\end{aligned}$$

整理得

$$|x^* - x_k| \leqslant \frac{1}{1-q}|x_{k+1} - x_k| \quad (k = 0,1,2,\cdots)$$

又因为

$$|x_{k+1} - x_k| = |\varphi(x_k) - \varphi(x_{k-1})| = |\varphi'(\xi_k)| \cdot |x_k - x_{k-1}| \leqslant q|x_k - x_{k-1}|$$

其中 ξ_k 在 x_k 和 x_{k-1} 之间。结合前面的推导，有

$$|x^* - x_k| \leqslant \frac{q}{1-q}|x_k - x_{k-1}| \leqslant \frac{q^2}{1-q}|x_{k-1} - x_{k-2}| \leqslant \cdots \leqslant \frac{q^k}{1-q}|x_1 - x_0|$$

根据定理，只要 $|x_{k+1} - x_k|$ 充分小，就可以保证迭代值足够精确，即满足 $|x_k - x^*| < \varepsilon$。所以常用 $|x_{k+1} - x_k| < \varepsilon'$ 来控制迭代过程的结束，并取方程根的近似值为 $x^* \approx x_k$。

定理既可以用来判断迭带函数是否收敛，还能估计迭代次数，但结果偏保守。若要求

$$|x^* - x_k| \leqslant \frac{q^k}{1-q}|x_1 - x_0| < \varepsilon$$

则迭代次数必须满足

$$k > \left| \ln \frac{\varepsilon(1-q)}{|x_1 - x_0|} \Big/ \ln q \right| \tag{6.15}$$

从几何上讲，求方程 $x = \varphi(x)$ 根的问题，就是求曲线 $y = \varphi(x)$ 和直线 $y = x$ 交点的横坐标 x^*。当迭代函数的导数 $\varphi'(x)$ 在根 x^* 附近取值范围不同时，迭代序列 $\{x_k\}$ 的收敛情况不同，如图 6.2 所示。图中(a) 和(b) 表示是收敛的，曲线 $y = \varphi(x)$ 比较平坦；(c) 和(d) 表示是发散的，曲线 $y = \varphi(x)$ 比较陡峭。

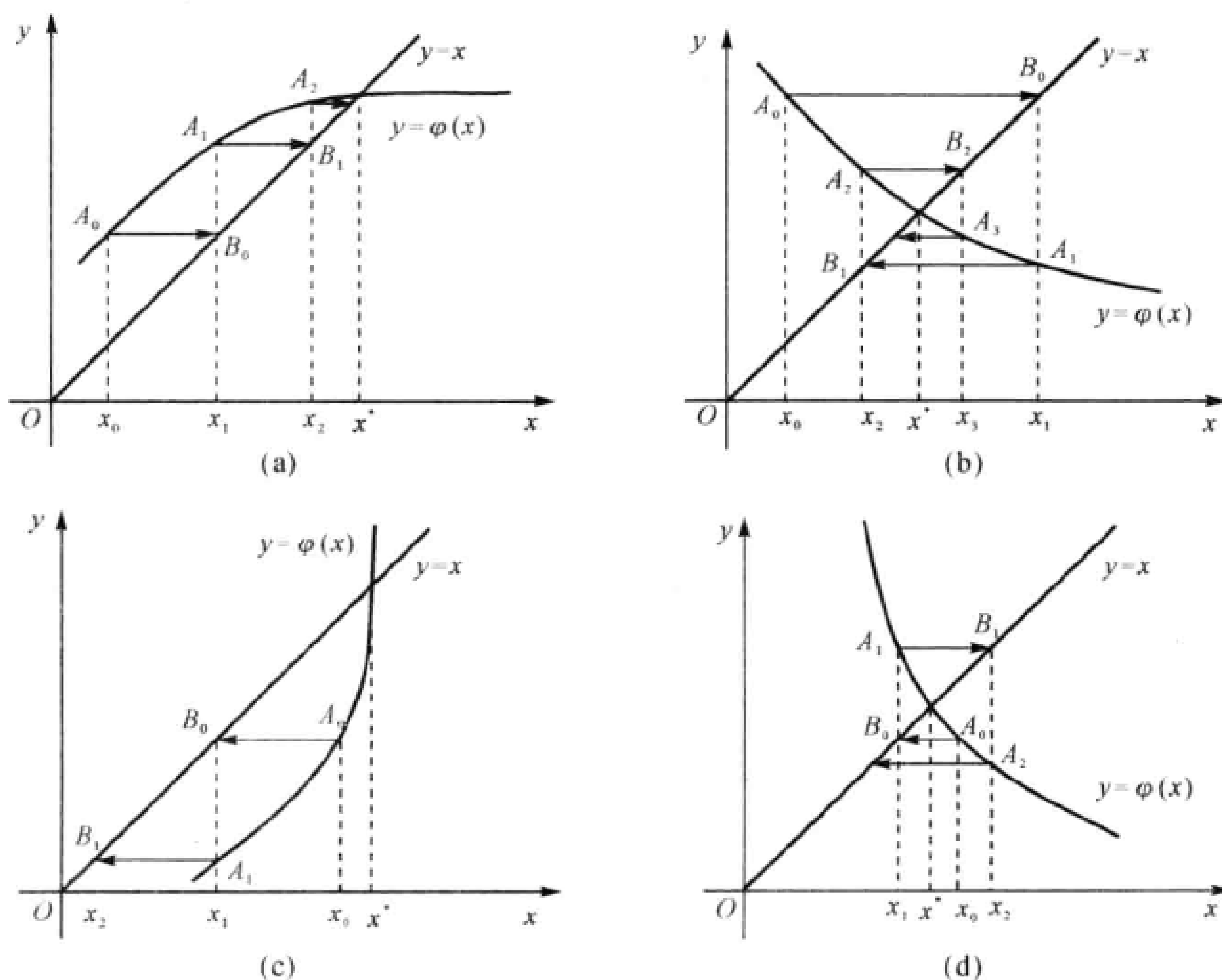


图 6.2

首先，从曲线 $y = \varphi(x)$ 上点 $A_0(x_0, \varphi(x_0))$ 出发，平行于 x 轴作直线 $y = \varphi(x_0)$，与直线 $y = x$ 相交于点 $B_0(x_1, x_1)$，其中 $x_1 = \varphi(x_0)$。再过点 $B_0(x_1, x_1)$ 作平行于 y 轴的直线 $x = x_1$，

与曲线 $y=\varphi(x)$ 相交于点 $A_1(x_1,\varphi(x_1))$；接着，从曲线 $y=\varphi(x)$ 上点 $A_1(x_1,\varphi(x_1))$ 出发，平行于 x 轴作直线 $y=\varphi(x_1)$，与直线 $y=x$ 相交于点 $B_1(x_2,x_2)$，其中 $x_2=\varphi(x_1)$。再过点 $B_1(x_2,x_2)$ 作平行于 y 轴的直线 $x=x_2$，与曲线 $y=\varphi(x)$ 相交于点 $A_2(x_2,\varphi(x_2))$；……；如此就能建立迭代序列 $\{x_k\}$。在 x 轴上，如果点序列 $x_0,x_1,\cdots,x_k,\cdots$ 向不动点 x^* 无限靠近，则迭代序列 $\{x_k\}$ 收敛于 x^*。否则，迭代序列 $\{x_k\}$ 不收敛。

例 6.2　方程 $x=e^{-x}$ 有唯一的实根 $x^*\in(0,1)$，迭代公式为 $x_{k+1}=e^{-x_k}(k=0,1,2,\cdots)$，讨论其收敛性，并求出满足条件 $|x_{k+1}-x_k|<10^{-3}$ 的方程根的近似值。

解　迭代函数为 $\varphi(x)=e^{-x}$，$\varphi'(x)=-e^{-x}$。显然，在区间 $(0,1)$ 内 $\varphi'(x)$ 连续并且 $|\varphi'(x)|=e^{-x}<e^0=1$。由上述定理可知，迭代公式 $x_{k+1}=e^{-x_k}(k=0,1,2,\cdots)$ 收敛。

取初始值 $x_0=0.5$，计算结果列于表 6.2 中。

表　6.2

k	x_k	x_k-x_{k-1}	k	x_k	x_k-x_{k-1}
1	0.606 53	0.106 53	6	0.564 86	−0.00 631
2	0.545 24	−0.061 29	7	0.568 44	0.003 58
3	0.579 70	0.034 46	8	0.566 41	−0.002 03
4	0.560 06	−0.019 64	9	0.567 56	0.001 15
5	0.571 17	0.011 11	10	0.566 91	−0.000 65

可见，迭代 10 次时结果满足 $|x_{k+1}-x_k|<10^{-3}$，所以取 $x^*\approx x_{10}=0.566\ 91$。

若有根区间 $[a,b]$ 较大，则收敛性定理给出的收敛条件 $|\varphi'(x)|\leqslant q<1$ 有时可能不成立。因此，实际迭代法中，常在根的小邻域内考虑问题。

定义(6.3)　如果在根的某个邻域 $B=\{x\mid |x-x^*|\leqslant\delta\}$ 中，对于任意的 $x_0\in B$，迭代公式 $x_{k+1}=\varphi(x_k)(k=0,1,2.\cdots)$ 均收敛，则称迭代公式在 x^* 附近局部收敛。

定理(6.3)　设 $x^*=\varphi(x^*)$，在 x^* 的某个邻域 $B=\{x\mid |x-x^*|\leqslant\delta\}$ 内 $\varphi'(x)$ 连续，并且 $|\varphi'(x)|\leqslant q<1$，则对于任何 $x_0\in B$，迭代公式 $x_{k+1}=\varphi(x_k)$ 建立的序列 $\{x_k\}$ 都收敛于 x^*，同时有

$$|x_k-x^*|\leqslant\frac{1}{1-q}|x_{k+1}-x_k|,\quad |x_k-x^*|\leqslant\frac{q^k}{1-q}|x_1-x_0|$$

由于方程的根 x^* 事先不知道，所以无法判断 $|\varphi'(x^*)|<1$ 是否成立。但如果函数 $\varphi'(x)$ 连续，则当有 $x_0\in B$ 使 $|\varphi'(x_0)|<1$ 时，就可以认为 $|\varphi'(x^*)|<1$ 也成立。

6.2.3　迭代法的收敛阶

人们总希望所采用的迭代方法不仅收敛，而且收敛得比较快，这就是迭代法的收敛速度问题。所谓迭代法的收敛速度，就是迭代序列在接近收敛时迭代误差 $e_k=x^*-x_k$ 的下降速度。为了描述迭代序列的收敛速度，引入收敛阶的概念。

定义(6.4)　设序列 $\{x_k\}$ 收敛于 x^*，令 $e_k=x^*-x_k$，若存在常数 $p\geqslant1$ 及正常数 C，使

$$\lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|^p}=C \tag{6.16}$$

则称序列$\{x_k\}$为p阶收敛。当$p=1$，且$C<1$时，称为线性收敛；当$p=2$时，称为平方收敛（或二次收敛）；当$p>1$时，称为超线性收敛。

如果迭代函数$\varphi(x)$产生的序列$\{x_k\}$是p阶收敛的，则称函数$\varphi(x)$是p阶迭代函数，并称迭代公式$x_{k+1}=\varphi(x_k)$是p阶收敛的。

定理(6.4) 设迭代函数$\varphi(x)$在方程$x=\varphi(x)$的根x^*邻近有连续的二阶导数，并且$|\varphi'(x)|<1$，则

(1) 当$\varphi'(x^*)\neq 0$时，迭代公式$x_{k+1}=\varphi(x_k)$为线性收敛；

(2) 当$\varphi'(x^*)=0$，但$\varphi''(x^*)\neq 0$时，迭代公式$x_{k+1}=\varphi(x_k)$为平方收敛。

(3) 一般地，若迭代函数$\varphi(x)$在方程$x=\varphi(x)$的根x^*邻近有连续的p阶导数，并且$\varphi'(x^*)=\varphi''(x^*)=\varphi'''(x^*)=\cdots=\varphi^{(p-1)}(x^*)=0$，而$\varphi^{(p)}(x^*)\neq 0$，则迭代公式$x_{k+1}=\varphi(x_k)$在$x^*$附近$p$阶收敛。

程序(6.2) 用迭代法计算方程根近似值的MATLAB程序。

程序任务：用迭代法计算方程$f(x)=0$的根的近似值。

```
function [x0, k] = RootIte(funiterate, xs, ep, Nm)
% 输入:funiterate—— 迭代函数 φ(x) (方程等价于 x = φ(x))
%        xs—— 迭代初值
%        ep—— 计算精度(连续两次迭代值差的绝对值小于ep时计算结束,并以最新迭代值作为根的近
%             似值)
%        Nm—— 最大迭代次数
% 输出:x0—— 根的近似值
%        k—— 迭代次数(输出 k = Nm 表示已经达到最大迭代次数,计算结果尚未达到精度要求,可以
%             适当增大 Nm,重新计算)
x = xs; xs = x + 2 * ep; k = 0;
while abs(xs - x) > ep & k < Nm
  k = k + 1;
  xs = x; x = feval(funiterate, xs);      % x_{k+1} = φ(x_k)
end
x0 = x;
if k == Nm warning('已经达到迭代次数上限'); end
```

用程序(6.2)计算方程$f(x)=x^3-x-1=0$在区间[1,1.5]内的根。

在MATLAB命令窗口输入：

```
>> fun = inline('(x + 1)^(1/3)');              % 定义迭代函数 φ(x) = ∛(x+1)
>> [x0, k] = RootIte(fun, 1.5, 1e-5, 50)       % 调用程序(6.2)计算
```

输出结果：

```
x0 = 1.3247
k = 7
```

即方程根的近似值是1.324 7，迭代次数为7。

6.3 牛顿迭代法

牛顿迭代法的基本思想是将非线性函数 $f(x)$ 逐步线性化，从而将非线性方程近似为线性方程来求解。

6.3.1 迭代公式

设方程 $f(x)=0$ 根 x^* 的某一近似值是 x_k，把函数 $f(x)$ 在 x_k 处展开成泰勒级数

$$f(x)=f(x_k)+f'(x_k)(x-x_k)+\frac{1}{2}f''(\xi)(x-x_k)^2$$

用前两项近似 $f(x)$（称为 $f(x)$ 的线性化），即用线性方程

$$f(x_k)+f'(x_k)(x-x_k)=0 \tag{6.17}$$

来近似方程 $f(x)=0$。若 $f'(x_k)\neq 0$，则用线性方程式(6.17)的根

$$x=x_k-\frac{f(x_k)}{f'(x_k)} \tag{6.18}$$

作为方程 $f(x)=0$ 根 x^* 的近似值。公式

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)} \quad (k=0,1,2,\cdots) \tag{6.19}$$

就是牛顿迭代法的迭代公式。

牛顿迭代法具有明显的几何意义。如图 6.3 所示，方程 $f(x)=0$ 的根就是曲线 $y=f(x)$ 与 x 轴的交点 x^*。用牛顿迭代法计算 x^* 近似值的过程是：过曲线 $y=f(x)$ 上点 $A_k(x_k,f(x_k))$ 作曲线的切线

$$y-f(x_k)=f'(x_k)(x-x_k)$$

其与 x 轴的交点坐标为 $x_{k+1}=x_k-f(x_k)/f'(x_k)$，它就是牛顿迭代公式(6.19)的计算结果；接着，再过曲线上点 $A_{k+1}(x_{k+1},f(x_{k+1}))$ 作曲线的切线，切线与 x 轴的交点坐标为 $x_{k+2}=x_{k+1}-f(x_{k+1})/f'(x_{k+1})$；……。从图上可以看出，只要迭代初值选取合适，迭代序列 $\{x_k\}$ 会很快收敛于方程精确解 x^*。由于牛顿迭代法在几何上的直观性，牛顿迭代法也被称为切线法。

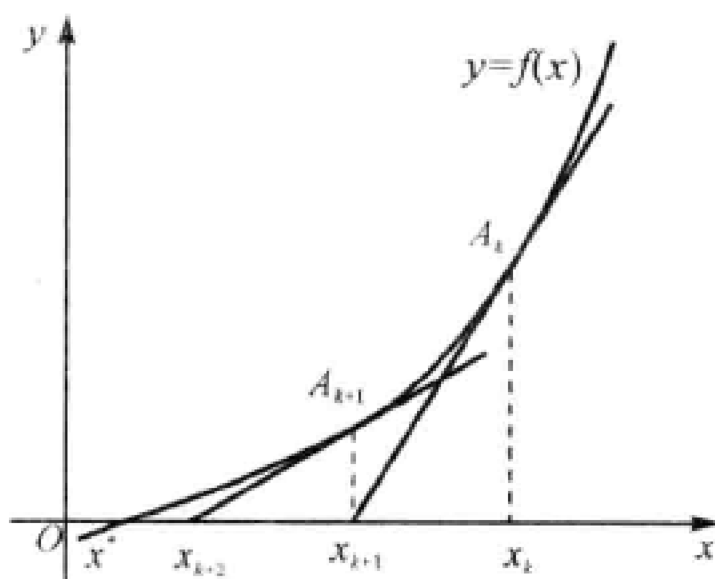


图　6.3

6.3.2 牛顿迭代法的收敛性

牛顿迭代法对应的方程是

$$x=x-\frac{f(x)}{f'(x)} \tag{6.20}$$

显然，它与 $f(x)=0$ 等价。牛顿迭代法的迭代函数为

$$\varphi(x)=x-\frac{f(x)}{f'(x)} \tag{6.21}$$

由于

$$\varphi'(x)=\frac{f(x)f''(x)}{[f'(x)]^2}$$

如果 x^* 是方程 $f(x)=0$ 的单根，即 $f(x^*)=0$，但 $f'(x^*)\neq 0$，则 $\varphi'(x^*)=0$。所以在单根附近，对于任意的初值，牛顿迭代序列都收敛于方程的根，即牛顿迭代法具有局部收敛性。并且可以证明，此时牛顿迭代法具有二阶收敛速度。但如果 x^* 是方程的重根，则牛顿迭代法仅有线性收敛速度，且根的重数越高收敛越慢。

定理(6.5) 设函数 $f(x)$ 在区间 $[a,b]$ 上有二阶导数，并且满足

(1) $f(a)\cdot f(b)<0$；

(2) $f'(x)\neq 0\ (x\in[a,b])$；

(3) $f''(x)$ 在 $[a,b]$ 上保持符号不变。

则在 $[a,b]$ 上任选满足条件 $f(x_0)\cdot f''(x_0)>0$ 的初始近似值 x_0，牛顿迭代法产生的迭代序列 $\{x_k\}$ 单调收敛于方程 $f(x)=0$ 在 $[a,b]$ 上的唯一根。

对定理(6.5)给出简单几何解释：条件(1)保证了方程 $f(x)=0$ 在 $[a,b]$ 上有根；条件(2)保证了函数 $f(x)$ 在 $[a,b]$ 上的单调性，因此方程有唯一根；条件(3)保证了函数 $f(x)$ 在 $[a,b]$ 上的凹凸方向不变化，即 $f'(x)$ 的单调性；条件 $f(x_0)\cdot f''(x_0)>0$ 保证了 $x_k\in[a,b]$ 时，有 $x_{k+1}=\varphi(x_k)\in[a,b]$。图 6.4 给出了 $f'(x)$ 和 $f''(x)$ 取不同符号的四种组合，考虑各种情况发现，只要初始近似值 x_0 满足条件 $f(x_0)\cdot f''(x_0)>0$，则迭代序列收敛于方程的根。

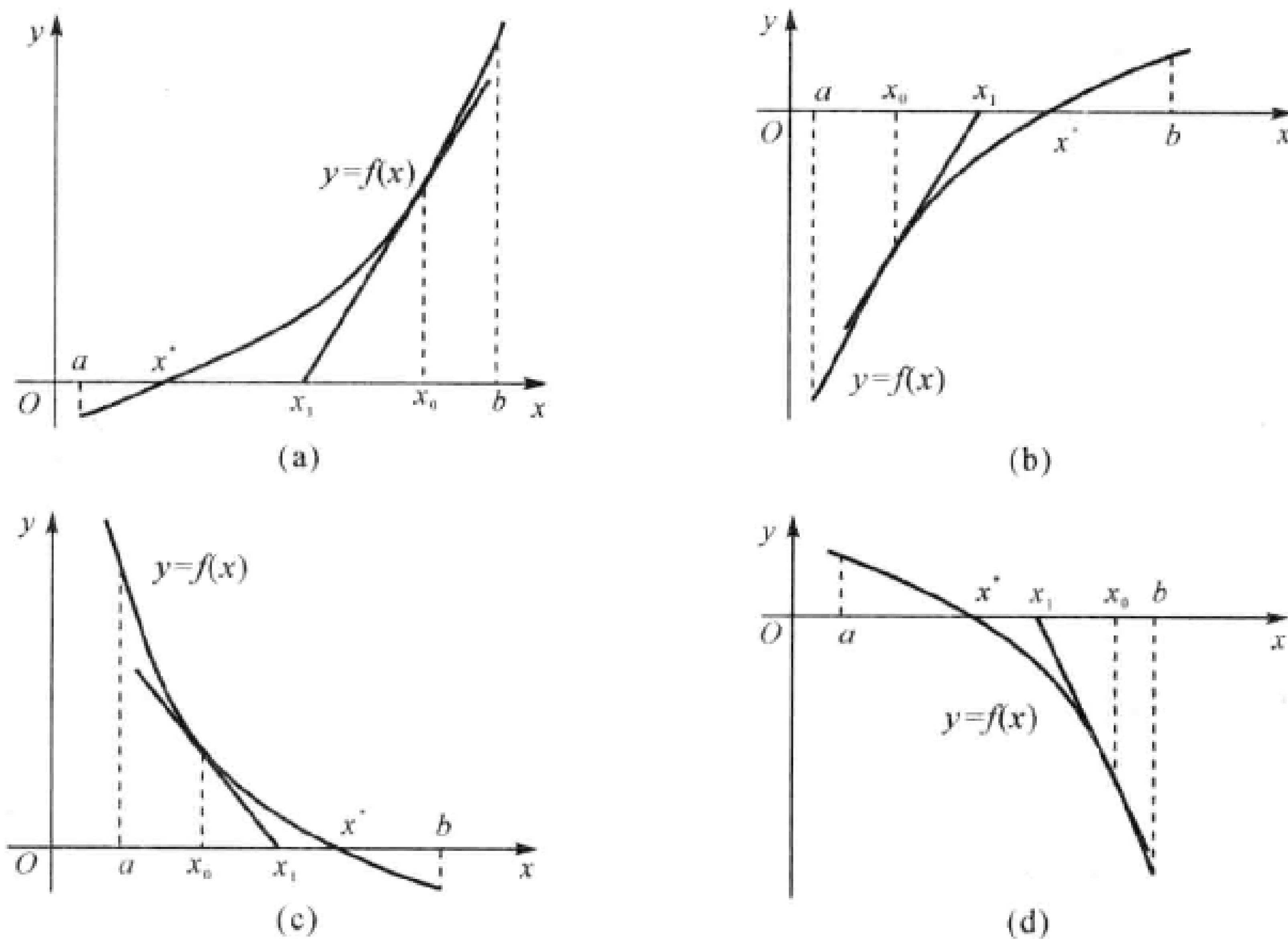


图 6.4

(a) $f'(x)>0, f''(x)>0$；(b) $f'(x)>0, f''(x)<0$；

(c) $f'(x)<0, f''(x)>0$；(d) $f'(x)<0, f''(x)<0$

例 6.3 对于给定的正数 a，用牛顿迭代法建立计算 $\sqrt{a}$ 的收敛迭代公式，并计算 $\sqrt{5}$ 的近似值，要求 $|x_{k+1}-x_k|<10^{-6}$。

解　令 $f(x)=x^2-a$，并且 $x>0$，则方程 $f(x)=0$ 的根就是 $\sqrt{a}$。用牛顿迭代法求解的迭代公式是

$$x_{k+1}=x_k-\frac{x_k^2-a}{2x_k}=\frac{1}{2}\left(x_k+\frac{a}{x_k}\right)\quad(k=0,1,2.\cdots)$$

当取 $x>0$ 时，$f'(x)=2x>0$，$f''(x)=2>0$。由定理(6.5)可知，对于满足 $x_0>\sqrt{a}$ 的任意初始近似值，形成的迭代序列都收敛于 $\sqrt{a}$ 的准确值。

取初始值 $x_0=2$，建立的迭代序列见表 6.3。

表　6.3

k	1	2	3	4
x_k	2.25	2.236 11	2.236 068	2.236 068
x_k-x_{k-1}	0.25	−0.013 89	−0.000 042	0.000 000

故满足要求的计算结果是 $\sqrt{5}\approx x_4=2.236\,068$。

程序(6.3)　用牛顿迭代法计算方程根近似值的 MATLAB 程序。

程序任务：用牛顿迭代法计算方程 $f(x)=0$ 的根的近似值。

```
function [x0, k] = RootNew(funew, dfun, xs, ep, Nm)
% 输入:funew—— 函数 f(x)
%      dfun—— 函数 f'(x)
%      xs—— 迭代初值
%      ep—— 计算精度(连续两次迭代值差的绝对值小于 ep 时计算结束,并以最新迭代值作为根的近
%           似值)
%      Nm—— 最大迭代次数
% 输出:x0—— 根的近似值
%      k—— 迭代次数(输出 k = Nm 表示已经达到最大迭代次数,但计算结果尚未达到精度要求,可
%          以适当增大 Nm,重新计算)
x = xs; xs = x + 2 * ep; k = 0;
while abs(xs - x) > ep & k < Nm
    k = k + 1;
    xs = x; x = x - feval(funew, xs)/ feval(dfun, xs);     % x_{k+1} = x_k - f(x_k) / f'(x_k)
end
x0 = x;
if k == Nm warning('已经达到迭代次数上限'); end
```

用程序(6.3)计算方程 $f(x)=x^3-3x-2=0$ 的根，取迭代初值为 3。

在 MATLAB 命令窗口输入：

```
>> funew = inline('x^3 - 3 * x - 2');                 % 定义函数 f(x)
>> dfun = inline('3 * x^2 - 3');                      % 定义函数 f'(x)
>> [x0, k] = RootNew('funew', 'dfun', 3, 1e-5, 50)    % 调用程序(6.3)计算
```

输出结果：

x0 =2.0000

k =5

即方程根的近似值是 2,迭代次数为 5。

6.4 弦 截 法

为了克服牛顿迭代法在迭代过程中不断计算函数导数值 $f'(x_k)$ 的明显不足,建立了弦截法。

在牛顿迭代公式(6.19)中用差商$\dfrac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$代替导数 $f'(x_k)$,就能得到弦截法迭代公式

$$x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_{k-1})}(x_k-x_{k-1})\quad (k=1,2,3.\cdots)\tag{6.22}$$

"弦截法"的名称由其几何意义得名。弦截法需要 x^* 邻近的两个初始近似值 x_0 和 x_1 才能进行迭代运算。该方法中方程 $f(x)=0$ 的等价形式是

$$x=x-\frac{f(x)}{f(x)-f(x_{k-1})}(x-x_{k-1})\tag{6.23}$$

相应的迭代函数是

$$\varphi(x)=x-\frac{f(x)}{f(x)-f(x_{k-1})}(x-x_{k-1})\tag{6.24}$$

用弦截法求方程根近似值,在几何上(见图 6.5)就是过点 $A(x_{k-1},f(x_{k-1}))$ 和点 $B(x_k,f(x_k))$ 作直线 AB,直线方程为

$$\frac{y-f(x_k)}{x-x_k}=\frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$$

用弦 AB 与 x 轴的交点坐标

$$x_{k+1}=x_k-\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}f(x_k)$$

作为方程 $f(x)=0$ 根的新近似值。

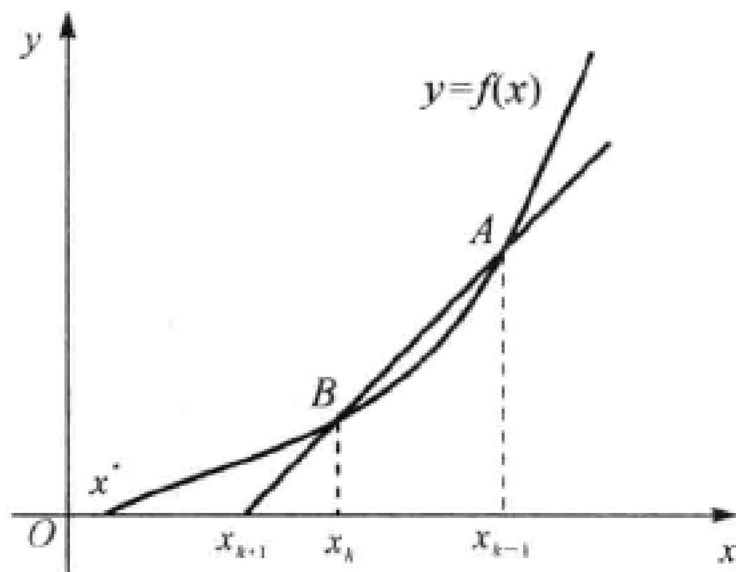


图 6.5

定理(6.6) 设方程 $f(x)=0$ 的根为 x^*,若 $f(x)$ 在 x^* 附近有连续的二阶导数,且 $f'(x)\neq 0$,则弦截法具有局部收敛性。当初始值 x_0 和 x_1 充分地接近 x^* 时,弦截法的迭代过程收敛,收敛速度为

$$|x_{k+1}-x^*|\approx\left|\frac{f''(x^*)}{2f'(x^*)}\right|^{0.618}|x_k-x^*|^{1.618}$$

即为超线性收敛的($p=1.618>1$)。

例 6.4 求方程 $f(x)=x^3-x^2-1=0$ 在 $x=1.5$ 邻近根的近似值 x_k,要求 $|f(x_k)|<\dfrac{1}{2}\times 10^{-7}$。

解 采用弦截法求解。根据式(6.22),建立迭代公式

$$x_{k+1}=x_k-\frac{x_k-x_{k-1}}{(x_k^3-x_k^2-1)-(x_{k-1}^3-x_{k-1}^2-1)}(x_k^3-x_k^2-1)=$$

$$x_k - \frac{x_k - x_{k-1}}{(x_k^3 - x_{k-1}^3) - (x_k^2 - x_{k-1}^2)}(x_k^3 - x_k^2 - 1)$$

取 $x_0 = 1.5$，$x_1 = 1.4$ 按上式进行迭代，计算结果列入表 6.4 中。

表　6.4

k	x_k	$\lvert f(x_k) \rvert$
2	1.463 343 108	7.8×10^{-3}
3	1.465 719 207	5.2×10^{-4}
4	1.465 570 923	1.1×10^{-6}
5	1.465 571 244	2.9×10^{-8}

求解结果为 $x^* \approx x_5 = 1.465\ 571\ 244$。

程序(6.4)　用弦截法计算方程根近似值的 MATLAB 程序。

程序任务：用弦截法计算方程 $f(x)=0$ 的根的近似值。

```
function [x0, k] = RootSecant(funsec, xf, xs, ep, Nm)
% 输入:funsec—— 函数 f(x)
%      xf—— 迭代的第一初值
%      xs—— 迭代的第二初值
%      ep—— 计算精度(连续两次迭代值差的绝对值小于 ep 时计算结束,并以最新迭代值作为根的近
%           似值)
%      Nm—— 最大迭代次数
% 输出:x0—— 根的近似值
%      k—— 迭代次数(输出 k = Nm 表示已经达到最大迭代次数,但计算结果尚未达到精度要求,可
%          以适当增大 Nm,重新计算)
k = 0; x = xf + 2 * ep;
while abs(x - xf) > ep & k < Nm
    k = k + 1;
    x = xs - (xs - xf) * feval(funsec,xs)/(feval(funsec,xs) - feval(funsec,xf));
                                          %x_{k+1} = x_k - (x_k - x_{k-1}) f(x_k) /[f(x_k) - f(x_{k-1})]
xf = xs; xs = x;
end
x0 = x;
if k == Nm warning('已经达到迭代次数上限'); end
```

用程序(6.4)计算方程 $f(x) = x^3 - x + 2 = 0$ 的根，取迭代初值为 $x_0 = -1.5$ 及 $x_1 = -1.52$。

在 MATLAB 命令窗口输入：

```
>> funsec=inline('x^3 - x + 2');                      % 定义函数 f(x)
>> [x0, k]=RootSecant('funsec', -1.5, -1.52, 1e-8, 50) % 调用程序(6.4)计算
```

输出结果：

```
x0 =-1.521 4
k =4
```

即方程根的近似值是 -1.5214,迭代次数为 4。

6.5 解非线性方程组的迭代法

考虑二元非线性方程组

$$\left.\begin{aligned} f_1(x,y)=0 \\ f_2(x,y)=0 \end{aligned}\right\} \tag{6.25}$$

方程组中的每一个方程都描述了 xy 平面上的一条曲线,方程组的解对应这两条曲线的交点 (p,q),这个点称为方程组的固定点。方程组式(6.25)可以变形成等价方程组

$$\left.\begin{aligned} x=g_1(x,y) \\ y=g_2(x,y) \end{aligned}\right\} \tag{6.26}$$

其迭代公式是

$$\left.\begin{aligned} p_{k+1}=g_1(p_k,q_k) \\ q_{k+1}=g_2(p_k,q_k) \end{aligned}\right\} \tag{6.27}$$

定理(6.7) 设式(6.26)中的两个函数及它们的一阶偏导数在包含点 (p,q) 的区域内连续,如果初始点 (p_0,q_0) 足够接近固定点 (p,q),并且

$$\left.\begin{aligned} \left|\frac{\partial g_1}{\partial x}(p,q)\right|+\left|\frac{\partial g_1}{\partial y}(p,q)\right|<1 \\ \left|\frac{\partial g_2}{\partial x}(p,q)\right|+\left|\frac{\partial g_2}{\partial y}(p,q)\right|<1 \end{aligned}\right\} \tag{6.28}$$

则式(6.27)的迭代将收敛到固定点 (p,q)。

可以根据式(6.27),结合解线性方程组的高斯-赛德尔迭代方法的思路,构造求解非线性方程组的赛德尔迭代公式

$$\left.\begin{aligned} p_{k+1}=g_1(p_k,q_k) \\ q_{k+1}=g_2(p_{k+1},q_k) \end{aligned}\quad (k=0,1,\cdots)\right\} \tag{6.29}$$

下面将解非线性方程的牛顿迭代法推广到求解非线性方程组。对于从 xy 平面到 uv 平面的变换

$$\left.\begin{aligned} u=f_1(x,y) \\ v=f_2(x,y) \end{aligned}\right\} \tag{6.30}$$

如果两个函数的偏导数连续,则在点 (x_0,y_0) 附近可以有下列近似

$$\left.\begin{aligned} u-u_0\approx\frac{\partial f_1}{\partial x}(x_0,y_0)(x-x_0)+\frac{\partial f_1}{\partial y}(x_0,y_0)(y-y_0) \\ v-v_0\approx\frac{\partial f_2}{\partial x}(x_0,y_0)(x-x_0)+\frac{\partial f_2}{\partial y}(x_0,y_0)(y-y_0) \end{aligned}\right\} \tag{6.31}$$

其中,$u_0=f_1(x_0,y_0)$,$v_0=f_2(x_0,y_0)$。上式代表一个局部线性变换,用矩阵乘法表示是

$$\Delta\boldsymbol{F}\approx\boldsymbol{J}(x_0,y_0)\Delta\boldsymbol{X} \tag{6.32}$$

式中,$\Delta\boldsymbol{F}=[u-u_0 \quad v-v_0]^{\mathrm{T}}$,$\Delta\boldsymbol{X}=[x-x_0 \quad y-y_0]^{\mathrm{T}}$,雅可比矩阵是

$$\boldsymbol{J}(x,y)=\begin{bmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{bmatrix} \tag{6.33}$$

方程组式(6.25)相当于在式(6.30)中令 $u=0,v=0$。设 (p,q) 是方程组式(6.25)的一组解,(p_0,q_0) 在 (p,q) 邻近,是方程组解的一个近似值,则函数在点 (p,q) 和点 (p_0,q_0) 处取值的差是

$$\left.\begin{aligned} u-u_0=f_1(p,q)-f_1(p_0,q_0)=0-f_1(p_0,q_0)\\ v-v_0=f_2(p,q)-f_2(p_0,q_0)=0-f_2(p_0,q_0)\end{aligned}\right\} \tag{6.34}$$

代入式(6.32)得到线性变换式

$$\boldsymbol{J}(p_0,q_0)\Delta\boldsymbol{P}\approx-\boldsymbol{F}(p_0,q_0) \tag{6.35}$$

其中

$$\boldsymbol{J}(p_0,q_0)=\begin{bmatrix}\dfrac{\partial f_1}{\partial x}(p_0,q_0) & \dfrac{\partial f_1}{\partial y}(p_0,q_0)\\ \dfrac{\partial f_2}{\partial x}(p_0,q_0) & \dfrac{\partial f_2}{\partial y}(p_0,q_0)\end{bmatrix},\quad \Delta\boldsymbol{P}=\begin{bmatrix}p-p_0\\ q-q_0\end{bmatrix},\quad \boldsymbol{F}(p_0,q_0)=\begin{bmatrix}f_1(p_0,q_0)\\ f_2(p_0,q_0)\end{bmatrix}$$

如果式(6.35)中的雅可比矩阵 $\boldsymbol{J}(p_0,q_0)$ 非奇异,则能解出

$$\Delta\boldsymbol{P}\approx-\boldsymbol{J}^{-1}(p_0,q_0)\boldsymbol{F}(p_0,q_0)$$

得到方程组解的又一个近似值

$$\boldsymbol{P}_1=\boldsymbol{P}_0+\Delta\boldsymbol{P}=\boldsymbol{P}_0-\boldsymbol{J}^{-1}(p_0,q_0)\boldsymbol{F}(p_0,q_0) \tag{6.36}$$

这个公式是牛顿迭代法公式(6.19)在二元方程组下的推广。因此,解非线性方程组的牛顿法的主要步骤如下:

(1) 计算函数值向量 $\boldsymbol{F}(\boldsymbol{p}_k)=\begin{bmatrix}f_1(p_k,q_k)\\ f_2(p_k,q_k)\end{bmatrix}$;

(2) 计算雅可比矩阵 $\boldsymbol{J}(\boldsymbol{p}_k)=\begin{bmatrix}\dfrac{\partial f_1}{\partial x}(p_k,q_k) & \dfrac{\partial f_1}{\partial y}(p_k,q_k)\\ \dfrac{\partial f_2}{\partial x}(p_k,q_k) & \dfrac{\partial f_2}{\partial y}(p_k,q_k)\end{bmatrix}$;

(3) 解线性方程组 $\boldsymbol{J}(\boldsymbol{p}_k)\Delta\boldsymbol{P}_k\approx-\boldsymbol{F}(\boldsymbol{p}_k)$,得出 $\Delta\boldsymbol{P}_k$;

(4) 计算下一近似值 $\boldsymbol{P}_{k+1}=\boldsymbol{P}_k+\Delta\boldsymbol{P}_k$。

6.6 物理学中的应用举例——平行共轴三线圈形成匀强磁场的条件

笔者曾提出用平行共轴的三个圆线圈产生匀强磁场的设想①。图 6.6 所示是平行共轴的三个圆线圈系统,三线圈中两侧线圈的半径均为 R,中间线圈的半径为 r',相邻两线圈之间的距离为 d,三个线圈中电流流向相同,电流强度均为 I。三线圈轴线上磁感应强度分布

$$B_z(z)=B_0\left\{\left[1+\left(\frac{z}{R}-\gamma\right)^2\right]^{-\frac{3}{2}}+\beta\left[1+\left(\frac{\beta z}{R}\right)^2\right]^{-\frac{3}{2}}+\left[1+\left(\frac{z}{R}+\gamma\right)^2\right]^{-\frac{3}{2}}\right\} \tag{6.37}$$

式中

$$B_0=\frac{\mu_0 I}{2R},\quad \beta=\frac{R}{r'},\quad \gamma=\frac{d}{R} \tag{6.38}$$

---

① 张引科,昝会萍. 共轴三线圈磁场的均匀性. 大学物理,2004,6(6):11-14.

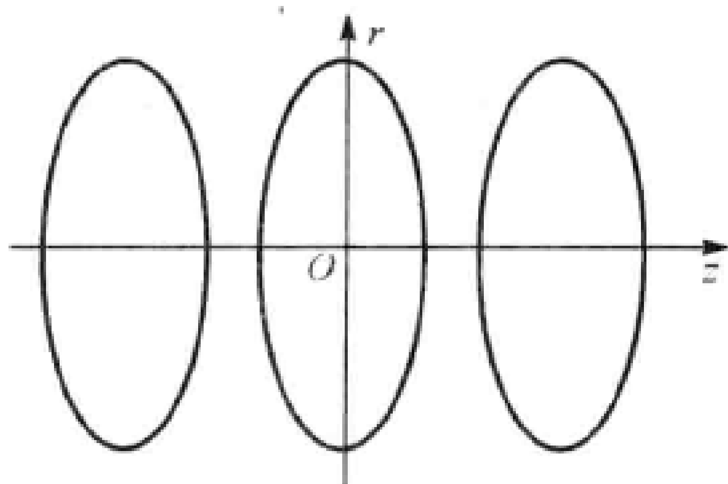


图 6.6

式(6.37)是变量 z 的偶函数。为了使函数值在原点附近区域的变化最小(磁场的均匀性最好),令式(6.37)对 z 的二阶导数和四阶导数在 $z=0$ 处的值等于零,得到方程

$$2\left(1+\gamma^{2}\right)^{-\frac{5}{2}}-10\gamma^{2}\left(1+\gamma^{2}\right)^{-\frac{7}{2}}+\beta^{3}=0 \tag{6.39}$$

$$-2\left(1+\gamma^{2}\right)^{-\frac{7}{2}}+28\gamma^{2}\left(1+\gamma^{2}\right)^{-\frac{9}{2}}-42\gamma^{4}\left(1+\gamma^{2}\right)^{-\frac{11}{2}}-\beta^{5}=0 \tag{6.40}$$

从方程式(6.40)中解出 β,代入方程式(6.39),得到关于 γ 的方程

$$\begin{aligned}f(\gamma)=&\left[-2\left(1+\gamma^{2}\right)^{-\frac{7}{2}}+28\gamma^{2}\left(1+\gamma^{2}\right)^{-\frac{9}{2}}-42\gamma^{4}\left(1+\gamma^{2}\right)^{-\frac{11}{2}}\right]^{3/5}+\\&2\left(1+\gamma^{2}\right)^{-\frac{5}{2}}-10\gamma^{2}\left(1+\gamma^{2}\right)^{-\frac{7}{2}}=0\end{aligned} \tag{6.41}$$

为了寻找方程式(6.41)根所在的区间,绘制 $f(\gamma)-\gamma$ 曲线,如图 6.7 所示。从图中可见,方程式(6.41)在[0.8,0.9]上存在唯一的根。调用程序(6.1)解方程式(6.41),得到其根是

$$\gamma=0.833\,42 \tag{6.42}$$

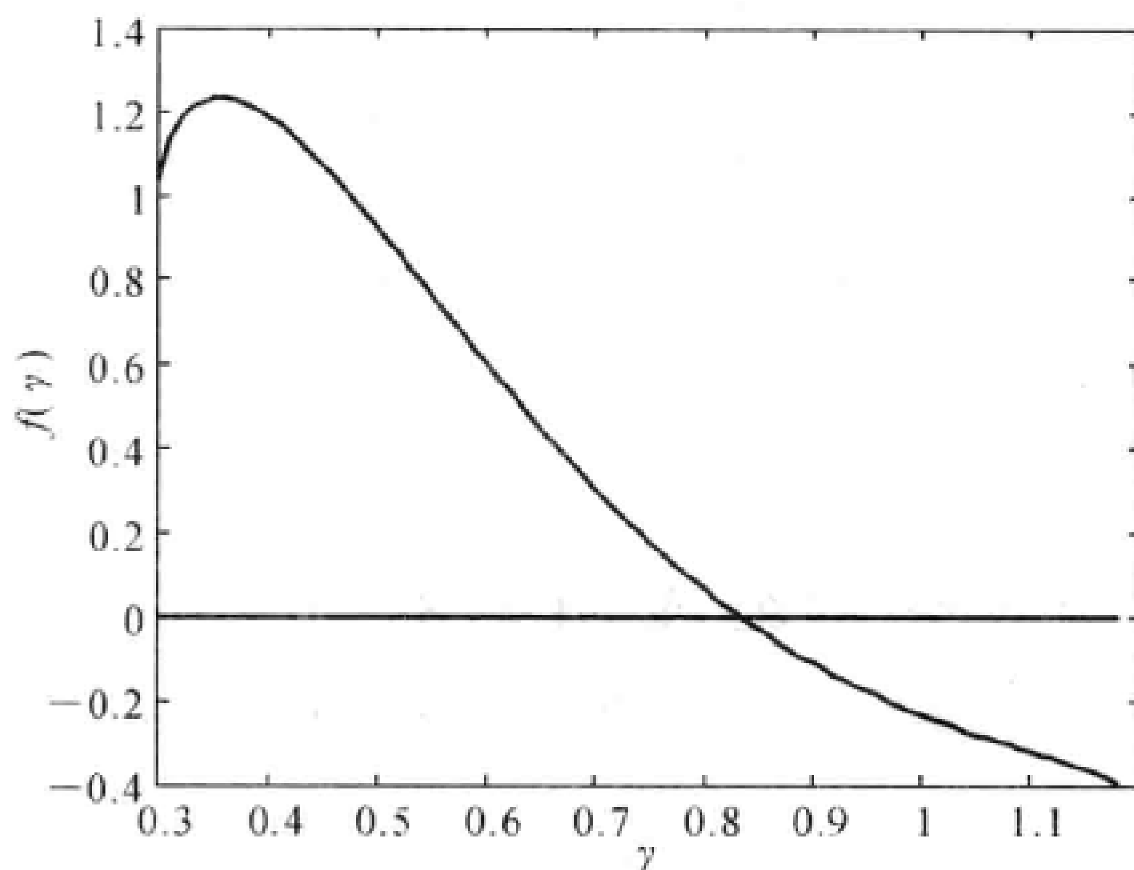


图 6.7

代入方程式(6.40),得到

$$\beta=0.824\,97 \tag{6.43}$$

由式(6.38)可得

$$r'=1.212\,16R,\quad d=0.833\,42R \tag{6.44}$$

这就是三线圈系统形成匀强磁场的条件。将 γ 和 β 的值代入公式(6.37),绘制的 $B_z/B_0-z/R$ 曲线如图 6.8 所示。图中实线表示三线圈系统轴线上的磁感应强度分布,虚线表示亥姆霍兹线圈轴线上的磁场分布。比较发现:① 在轴线上 R 范围内,亥姆霍兹线圈磁场起伏的相对误

差大于 5%，而三线圈系统磁场起伏的相对相对误差小于 0.5%；② 三线圈系统的磁场比亥姆霍兹线圈的磁场明显要强，中心位置处前者是后者的 1.210 倍。

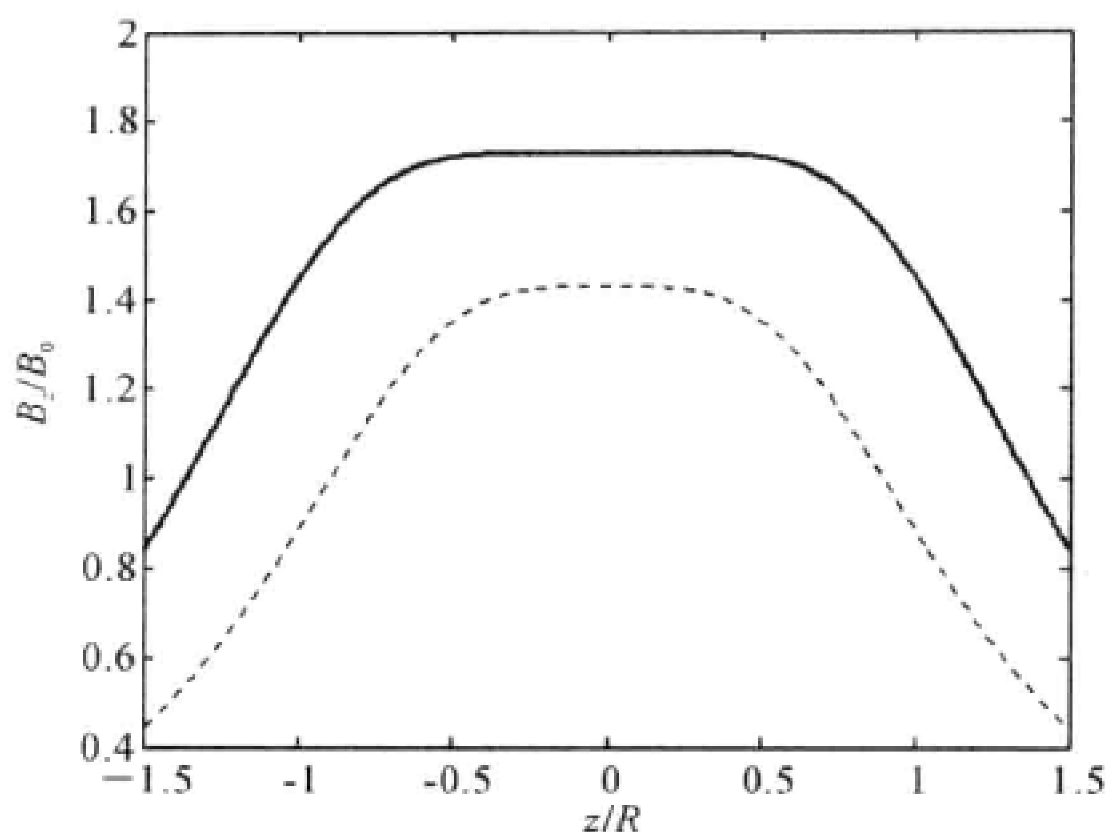


图　6.8

习　题

1. 用对分法求方程 $f(x)=x^3+4x^2-10=0$ 在区间[1,2]内的一个实根，要求满足精度 $|x^*-x_k|<10^{-3}$；若要求 $|x^*-x_k|<10^{-5}$，问需对分区间多少次？

2. 用迭代法求方程 $f(x)=x^3+4x^2-10=0$ 在区间[1,2]内的一个近似根，取初始近似值 $x_0=1.5$。

3. 用牛顿迭代法求方程 $f(x)=x^3+4x^2-10=0$ 在区间[1,2]内的一个实根，取初始近似值 $x_0=1.5$。

4. 用弦截法求方程 $f(x)=x^3-3x-1=0$ 在 2 附近的根。取 $x_0=2, x_1=1.9$，计算结果精确到 4 位有效数字。

5. 为了求解方程 $x^3-x^2-1=0$ 在 $x_0=1.5$ 附近的根，建立了三种不同的等价方程，并构造了相应的根近似值迭代公式：

(1) $x=1+\dfrac{1}{x^2}$，　　$x_{k+1}=1+\dfrac{1}{x_k^2}$；

(2) $x=\sqrt[3]{1+x^2}$，　　$x_{k+1}=\sqrt[3]{1+x_k^2}$；

(3) $x=\dfrac{1}{\sqrt{x-1}}$，　　$x_{k+1}=\dfrac{1}{\sqrt{x_k-1}}$。　$(k=0,1,2,\cdots)$

试判断各个迭代公式的收敛性，并估计其收敛阶次。选择一种收敛的迭代公式，计算方程根具有四位有效数字的近似值。

6. 用二分法和牛顿迭代法求方程 $x=\tan x$ 的最小正根。

7. 用牛顿法求解方程组 $\begin{cases}x^3+y^2-1=0\\x^3-y=0\end{cases}$ 在 $x_0=0.8$ 与 $y_0=0.6$ 附近的解，迭代 3 次。

8. 相距 100m 的两个高度相同的电塔之间悬挂一电缆，允许电缆中间下垂 10m，电缆形成

的曲线方程是

$$y = a\operatorname{ch}\frac{x}{a} \quad (x \in [-50,50])$$

(1) 用数值方法解非线性方程

$$a + 10 = a\operatorname{ch}\frac{50}{a}$$

确定参数 a；

(2) 用数值积分方法计算电缆的长度。

9. 考虑空气阻力作用时，投射体的飞行高度和水平飞行距离分别是

$$y = (Cv_0\sin\theta_0 + 32C^2)\left[1 - \exp\left(-\frac{t}{C}\right)\right] - 32Ct$$

$$x = Cv_0\cos\theta_0\left[1 - \exp\left(-\frac{t}{C}\right)\right]$$

式中，t 是飞行时间，v_0 是投射初速度，θ_0 是投射仰角，$C = m/k$，k 是空气阻力系数，m 是投射体的质量。若在水平地面上投射一物体，并且 $\theta_0 = \pi/6$，$v_0 = 200\text{m/s}$ 及 $C = 10\text{s}$，求投射体的飞行时间和飞行的水平距离。

10. 在我阵地前方 1 000m 处有一座高 50m 的山丘，上面建有一座敌方碉堡，那么我方大炮在什么角度下以最小速度发射炮弹就能摧毁敌方这座碉堡？炮弹的发射速度是多少？ 用数值积分方法计算炮弹飞行的路程。

11. 用迭代法计算二氧化碳气体的体积 V（要求精度 $\varepsilon = 10^{-4}$）。已知二氧化碳气体的状态方程是

$$\left(p + \frac{a}{V^2}\right)(V - b) = RT$$

其中，$a = 3.592$，$b = 0.042\,67$，$p = 200$，$T = 500$，$R = 0.082\,054$。

12. 跳伞运动员的下降速度由下面公式表示：

$$v = \frac{mg}{c}\left[1 - \exp\left(-\frac{c}{m}t\right)\right]$$

式中，g 是重力加速度（$\approx 10\text{m/s}^2$），m 是运动员的质量，c 是拽拉系数（$\approx 14\text{kg/s}$）。测得在下落 7s 时某运动员的速度为 35m/s，计算运动员的质量（要求误差不大于 0.01%）。

13. 圆孔和狭缝夫琅禾费衍射图样的强度分布分别为

$$I(r) = I(0)\left[\frac{2\mathrm{J}_1(kdr/2z)}{kdr/2z}\right]^2, \quad I'(x) = I'(0)\left[\frac{\sin(kwx/2z)}{kwx/2z}\right]^2$$

其中，d 是圆孔直径，w 是狭缝宽度，z 是观察屏到衍射屏的距离，$k = 2\pi/\lambda$，λ 是光波波长，r 是观察屏上极坐标系的径向坐标，x 是观察屏上直角坐标系的一个坐标，$\mathrm{J}_1(y)$ 是一阶第一类贝塞尔函数。

(1) 求圆孔衍射中央亮斑（爱里斑）中 $I(r)/I(0) = 1/2$ 的点到中心的距离 r_h，并计算 r_h 与爱里斑半径 $\Delta r = 1.22\lambda z/d$ 的比值；

(2) 求单缝衍射中央亮纹中 $I'(x)/I'(0) = 1/2$ 的点到中心的距离 x_h，并计算 x_h 与中央亮纹半宽度 $\Delta x = \lambda z/w$ 的比值；

(3) 建立圆孔衍射和单缝衍射一级亮纹中心坐标满足的方程，并计算中心坐标的近似值。

第7章　常微分方程的数值解法

科学研究和工程领域的许多过程，在数学上常用微分方程（或微分方程组）来描述，但除了少数特殊类型的微分方程以外，大多数微分方程是没有解析解的，或者求解其解析解十分困难。因此用数值方法寻找微分方程的近似解就很有必要。

求解微分方程近似解的方法分为两类：一类是近似解析法，如逐次逼近法、级数法等；另一类是数值解法，它可以求出方程精确解在一些离散点上的近似值。本章介绍一阶常微分方程的几种常用数值解法，如欧拉方法、龙格-库塔方法等，并讨论这些方法的收敛性和稳定性，最后简要介绍常微分方程组与高阶常微分方程的数值求解方法。

7.1　数值方法概述

只有结合定解条件，才能求出微分方程的具体解，微分方程和定解条件一起构成定解问题。定解条件有两种，一种是给出积分曲线在初始点的状态，称为初始条件，相应的定解问题称为初值问题；另一种是给出积分曲线首尾两端的状态，称为边界条件，相应的定解问题称为边值问题。

一阶常微分方程的初值问题是

$$\left.\begin{aligned} y' &= f(x,y) \\ y(x_0) &= y_0 \end{aligned}\quad (a \leqslant x \leqslant b)\right\} \tag{7.1}$$

本初值问题解的存在条件由下面定理给出。

定理(7.1)　设函数 $f(x,y)$ 在区域 $D=\{(x,y)\mid a \leqslant x \leqslant b, -\infty < y < +\infty\}$ 上连续，且满足李普希兹(Lipschitz)条件，即存在正数 L，使得对于 D 内任意两点 (x,y_1) 和 (x,y_2)，恒有

$$|f(x,y_1)-f(x,y_2)| \leqslant L|y_1-y_2| \tag{7.2}$$

则初值问题式(7.1)的解 $y(x)$ 存在且唯一。

数值方法的任务就是寻求初值问题这个唯一存在的解 $y(x)$ 在一些离散节点上的近似值。数值方法的基本思路是：在初值问题解存在的区间 $[a,b]$ 上插入一系列节点 $a=x_0<x_1<\cdots<x_{n-1}<x_n=b$，称 $h_i=x_i-x_{i-1}$ 为由节点 x_{i-1} 到节点 x_i 的步长。当步长 h_i 相等时，称为等步长，记作 $h=(b-a)/n$。在这些节点上用离散化方法（如数值积分、数值微分、泰勒级数展开等）将连续型的微分方程转化为离散型的代数方程（即差分方程）来求解。具体方法是：利用已知的初始值 $y_0=y(x_0)$，由具体算式求出下一节点处精确解 $y(x_1)$ 的近似值 y_1；再由 y_0 和 y_1 求出 $y(x_2)$ 的近似值 y_2；如此反复，直到求出 $y(x_n)$ 的近似值 y_n 为止。这种按节点排列顺序一步一步向前推进的方法称为步进式算法或递推式算法，所用的数值计算方法称为数值解法，求出的近似值 $y_i(i=0,1,\cdots,n)$ 称初值问题的数值解。如果在计算 y_k 时只用到前一节点 x_{k-1} 处的值 y_{k-1}，则步进式算法称为单步法。

当 $h\to 0$ 时，如果数值解 $y_i\ (i=0,1,\cdots,n)$ 趋向于准确解 $y(x_i)\ (i=0,1,\cdots,n)$，则称该方法是收敛的，否则是发散的。

7.2 欧拉方法

欧拉(Euler)方法是解常微分方程的一种最简单方法。由于其精度较低，实际计算中很少采用。但它最能体现数值解法的基本思想，因此很有必要进行介绍。

7.2.1 欧拉方法

定义(7.1) 若将函数 $y(x)$ 在点 x_i 处的导数 $y'(x_i)$ 用向前的一阶两点式代替，即

$$y'(x_i)\approx\frac{y_{i+1}-y_i}{h}$$

同时用 y_i 近似右端函数中的 $y(x_i)$，则初值问题式(7.1)转化为

$$\left.\begin{aligned}&y_{i+1}=y_i+hf(x_i,y_i)\\&y_0=y(x_0)\end{aligned}\quad(i=0,1,\cdots,n-1)\right\}\tag{7.3}$$

式(7.3)就是欧拉公式。利用它可以由 y_0 出发，计算出 y_1，再由 y_1 计算出 y_2，……。这种方法就称为欧拉方法。

欧拉方法有明确的几何意义。如图7.1所示，方程 $y'=f(x,y)$ 的解 $y=y(x)$ 是平面上的一族积分曲线。初值问题式(7.1)的解就是过点 (x_0,y_0) 的那条特殊积分曲线。欧拉方法的求解过程：首先，过初始点 $P_0(x_0,y_0)$ 作积分曲线 $y=y(x)$ 的切线(切线斜率为 $f(x_0,y_0)$)，该切线与直线 $x=x_1$ 相交于点 $P_1(x_1,y_1)$；然后，过点 P_1 作斜率为 $f(x_1,y_1)$ 的直线，该直线与直线 $x=x_2$ 相交于点 $P_2(x_2,y_2)$；……。如此继续，就能形成一条折线 $\overline{P_0P_1P_2\cdots P_n}$。可以证明，折线各个折点的纵坐标 $y_i\ (i=0,1,\cdots,n)$ 就是欧拉公式(7.3)计算的数值解。故此，欧拉方法也称为欧拉折线法。

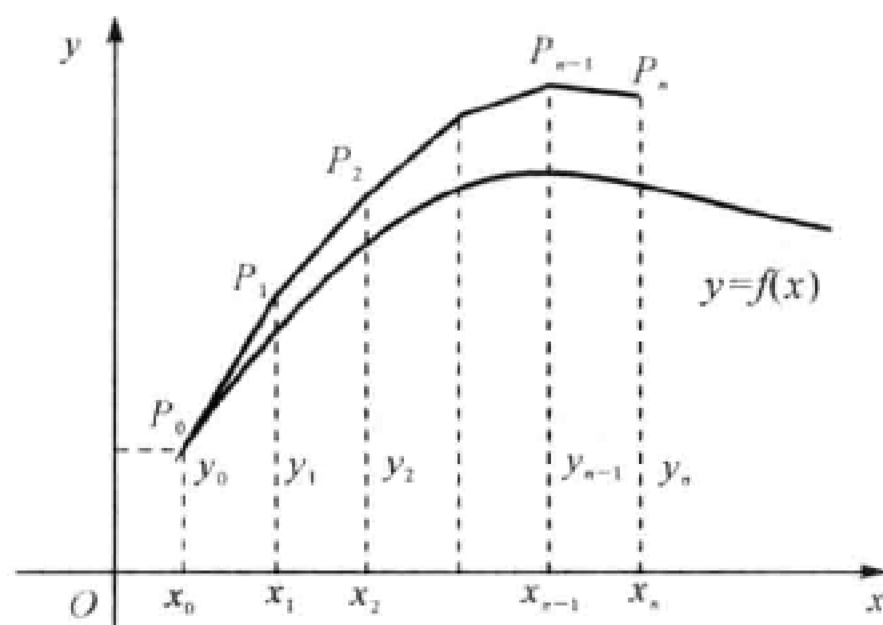


图 7.1

欧拉公式的其他两种推导方法：

(1) 将函数 $y(x)$ 在节点 x_i 处展开为泰勒级数，并计算

$$y(x_{i+1})=y(x_i)+hy'(x_i)+\frac{h^2}{2!}y''(x_i)+\cdots=y(x_i)+hf(x_i,y(x_i))+\frac{h^2}{2!}y''(x_i)+\cdots$$

保留至 h 的线性部分，并用 y_i 近似代替 $y(x_i)$，得

$$y_{i+1}=y_i+hf(x_i,y_i)$$

(2) 对方程 $\frac{dy}{dx}=f(x,y)$ 两边从 x_i 到 x_{i+1} 积分，有

$$y(x_{i+1})=y(x_i)+\int_{x_i}^{x_{i+1}}f(x,y(x))\,dx \tag{7.4}$$

用左矩形公式计算积分，即

$$\int_{x_i}^{x_{i+1}}f(x,y(x))\,dx\approx hf(x_i,y(x_i))$$

并用 y_i 近似代替 $y(x_i)$，得

$$y_{i+1}=y_i+hf(x_i,y_i)$$

故欧拉方法也称为矩形法。

程序(7.1)　用欧拉方法解微分方程初值问题的 MATLAB 程序。

程序任务：用欧拉公式(7.3)解初值问题式(7.1)。

```
function S = DeqEuler(dfun, a, b, y0, Nm)
% 输入:dfun—— 微分方程中的函数 f(x,y)
%      a,b—— 求解区间两端的坐标
%      y0—— 解函数 y(x) 的初始值 y0
%      Nm—— 区间[a, b] 被等分的数目
% 输出:S——S=[x' y'],其中 x 是数值解的节点坐标向量(1×(Nm+1)),y 是解函数在节点处的值向量(1×(Nm+1))
h = (b-a)/Nm;                % 相邻两个节点之间的距离
x = a:h:b;                   % 节点坐标向量
y = zeros(1, Nm+1);          % 初始化解函数值向量
y(1) = y0;                   % 赋初始值
for i = 1:Nm
    y(i+1) = y(i) + h*feval(dfun, x(i), y(i));    % 计算函数值 y(x_{i+1})
end
S = [x' y'];
```

用程序(7.1)解初值问题 $\begin{cases}y'=y-\dfrac{2x}{y}\\ y(0)=1\end{cases}$，取步长 $h=0.05$，求解区间为[0,0.5]。

在 MATLAB 命令窗口输入：

```
>> x=0:0.05:0.5;                              % 建立节点向量
>> ya=sqrt(1+2.*x);                           % 计算精确解函数在节点处的值向量
>> dfun=inline('y-2*x/y','x','y');            % 定义函数 f(x,y)=y-2x/y
>> S=DeqEuler(dfun, 0, 0.5, 1, 10);           % 调用程序(7.1)求解(Nm=10)
>> [S ya']                      % 输出:第一列是 x',第二列是 yi',第三列是精确解
```

输出结果见表 7.1。

表 7.1

i	x_i	数值解 y_i	精确解 $\sqrt{1+2x_i}$	i	x_i	数值解 y_i	精确解 $\sqrt{1+2x_i}$
1	0.050 0	1.050 0	1.048 8	6	0.300 0	1.271 3	1.264 9
2	0.100 0	1.097 7	1.095 4	7	0.350 0	1.311 3	1.303 8
3	0.150 0	1.143 5	1.140 2	8	0.400 0	1.350 1	1.341 6
4	0.200 0	1.187 6	1.183 2	9	0.450 0	1.388 0	1.378 4
5	0.250 0	1.230 1	1.224 7	10	0.500 0	1.425 0	1.414 2

7.2.2 梯形公式

若用梯形积分公式计算式(7.4)中的积分,即

$$\int_{x_i}^{x_{i+1}} f(x,y(x))\,\mathrm{d}x \approx \frac{h}{2}[f(x_i,y(x_i))+f(x_{i+1},y(x_{i+1}))]$$

并用 y_i 近似代替 $y(x_i)$,得到梯形公式

$$y_{i+1}=y_i+\frac{h}{2}[f(x_i,y_i)+f(x_{i+1},y_{i+1})] \tag{7.5}$$

欧拉公式(7.3)中的 y_{i+1} 可以直接由 y_i 求出,故称欧拉公式为显式方法。梯形公式(7.5)的右端含有未知的 y_{i+1},使得梯形公式成为关于 y_{i+1} 的方程,通常使用迭代法求解,所以把这类情形称为隐式方法。

梯形公式中的 y_{i+1} 要用迭代法求解。可先用欧拉公式算出迭代初值

$$y_{i+1}^{(0)}=y_i+hf(x_i,y_i) \tag{7.6}$$

然后用梯形公式进行迭代,即

$$y_{i+1}^{(k+1)}=y_i+\frac{h}{2}[f(x_i,y_i)+f(x_{i+1},y_{i+1}^{(k)})] \quad (k=0,1,2,\cdots) \tag{7.7}$$

迭代过程进行到连续两次计算结果之差小于给定精度,即

$$|y_{i+1}^{(k+1)}-y_{i+1}^{(k)}|<\varepsilon$$

为止,并取 $y_{i+1}=y_{i+1}^{(k+1)}$。然后转入下一步计算 y_{i+2}。

程序(7.2) 用梯形公式解微分方程初值问题的MATLAB程序。

程序任务:用梯形公式(7.5)解初值问题式(7.1)。

```
function S = DeqTrap(dfun, a, b, y0, Nm, ep)
% 输入:dfun—— 微分方程中的函数 f(x,y)
%       a,b—— 求解区间两端的坐标
%       y0—— 解函数 y(x) 的初始值 y0
%       Nm—— 区间[a, b]被等分的数目
%       ep—— 迭代法计算 yi 的精度
% 输出:S —— S = [x' y'],其中 x 是数值解的节点坐标向量(1×(Nm+1)),y 是解函数在节点处的值
%             向量(1×(Nm+1))
h = (b-a)/Nm;              % 相邻两个节点之间的距离
```

```
x = a:h:b;                       % 节点坐标向量
y = zeros(1, Nm + 1);            % 初始化解函数值向量
y(1) = y0;                       % 赋初始值
for i = 1:Nm
    y0 = y(i) + h * feval(dfun, x(i), y(i));      % 按照式(7.6)计算 y_{i+1}^{(0)}
    y1 = y0 + 2 * ep;
    j = 0;                                        % 迭代次数控制参数
    while abs(y1 - y0) > ep                       % 迭代精度控制,|y_{i+1}^{(k+1)} - y_{i+1}^{(k)}|< ε 时迭代结束
        j = j + 1;
        if j > 1e+3                               % 迭代次数大于 10^3 时结束,增大后重新计算
            break;
        end
        y1 = y0;                                  %yi 存放 y_{i+1}^{(k)}
        y0 = y(i) + (h/2) * (feval(dfun, x(i), y(i)) + feval(dfun, x(i + 1), y0));     % 计算 y_{i+1}^{(k+1)}
    end
    y(i + 1) = y0;                                % 存放 y_{i+1}
end
S = [x' y'];                                      % 输出计算结果[x' y']
```

用程序(7.2)解初值问题$\begin{cases} y' = y - \dfrac{2x}{y} \\ y(0) = 1 \end{cases}$,取步长 $h=0.1$,求解区间为[0, 1]。

在 MATLAB 命令窗口输入:

```
>> x=0:0.1:1.0;                              % 建立节点向量
>> ya=sqrt(1 + 2. * x);                      % 计算精确解函数在节点处的值向量
>> dfun=inline('y - 2 * x/y','x','y');       % 定义函数 f(x,y)=y-2x/y
>> S=DeqTrap(dfun, 0, 1, 1, 10, 1e-7);% 调用程序(7.2)求解
>> [S ya']       % 输出:第一列是 x',第二列是数值解 y',第三列是精确解 ya'
```

输出结果见表 7.2。

表　7.2

i	x_i	数值解 y_i	精确解 $\sqrt{1+2x_i}$	i	x_i	数值解 y_i	精确解 $\sqrt{1+2x_i}$
1	0.100 0	1.095 7	1.095 4	6	0.600 0	1.484 3	1.483 2
2	0.200 0	1.183 6	1.183 2	7	0.700 0	1.550 4	1.549 2
3	0.300 0	1.265 4	1.264 9	8	0.800 0	1.613 9	1.612 5
4	0.400 0	1.342 3	1.341 6	9	0.900 0	1.675 1	1.673 3
5	0.500 0	1.415 1	1.414 2	10	1.000 0	1.734 1	1.732 1

7.2.3　预估-校正法

梯形公式中的 y_{i+1} 要用迭代法求解,增加了计算量。为了保持梯形公式精度高的优点,同时克服其需要迭代运算的缺点,在实际计算中将欧拉公式和梯形公式结合使用,形成了预估-

校正法。具体作法如下：

(1) 先用欧拉公式求出 y_{i+1} 的一个初步近似值，记作 $\bar{y}_{i+1}$，称之为预估值，即

$$\bar{y}_{i+1} = y_i + hf(x_i, y_i) \tag{7.8}$$

(2) 再用梯形公式对预估值进行校正，即用预估值 $\bar{y}_{i+1}$ 代替梯形公式(7.5)右端函数中的 y_{i+1} 进行计算，得到的 y_{i+1} 就是校正值，即

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, \bar{y}_{i+1})] \tag{7.9}$$

两步结合起来就是

$$\left.\begin{aligned} \bar{y}_{i+1} &= y_i + hf(x_i, y_i) \\ y_{i+1} &= y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, \bar{y}_{i+1})] \end{aligned} \quad (i = 0, 1, \cdots, n-1) \right\} \tag{7.10}$$

这样建立的预估-校正公式也称为改进的欧拉公式。

为了便于编程，常把预估-校正公式(7.10)写成下列形式：

$$\left.\begin{aligned} y_{i+1} &= y_i + \frac{1}{2}(k_1 + k_2) \\ k_1 &= hf(x_i, y_i) \\ k_2 &= hf(x_i + h, y_i + k_1) \end{aligned} \quad (i = 0, 1, \cdots, n-1) \right\} \tag{7.11}$$

程序(7.3) 用预估-校正公式解微分方程初值问题的 MATLAB 程序。

程序任务：用预估-校正公式(7.11)解初值问题(7.1)。

```
function S = DeqEuler2(dfun, a, b, y0, Nm)
% 输入:dfun—— 微分方程中的函数 f(x,y)
%        a,b—— 求解区间两端的坐标
%        y0—— 解函数 y(x) 的初始值 y0
%        Nm—— 区间[a, b]被等分的数目
% 输出:S—— S = [x' y'],其中 x 是数值解的节点坐标向量(1×(Nm+1)),y 是解函数在节点处的值
%              向量(1×(Nm+1))
h = (b - a)/Nm;              % 相邻两个节点之间的距离
x = a:h:b;                   % 节点坐标向量
y = zeros(1, Nm + 1);        % 初始化解函数值向量
y(1) = y0;                   % 赋初始值
for i = 1:Nm
    k1 = h * feval(dfun, x(i), y(i));          % 按照式(7.11)第二式计算 k1
    k2 = h * feval(dfun, x(i) + h, y(i) + k1); % 按照式(7.11)第三式式计算 k2
    y(i + 1) = y(i) + (k1 + k2)/2;             % 按照式(7.11)第一式计算 y_{i+1}
end
S = [x' y'];                                   % 输出计算结果[x' y']
```

用程序(7.3)解初值问题 $\begin{cases} y' = e^{-2x} - 2y \\ y(0) = 0.1 \end{cases}$，取步长 $h = 0.05$，求解区间为[0, 1]。

在 MATLAB 命令窗口输入：

```
>> x=0:0.05:1.0;                          % 建立节点向量
```

```
>> ya=exp(-2.*x)./10+x.*exp(-2.*x);  % 计算精确解函数在节点处的值向量
>> dfun=inline('exp(-2*x)-2*y','x','y');  % 定义函数 f(x,y)=e^(-2x)-2y
>> S=DeqEuler2(dfun,0,1,0.1,20);       % 调用程序(7.3)求解(Nm=20)
>> [S ya']    % 输出:第一列是 x',第二列是数值解 y',第三列是精确解 ya'
```

输出结果(部分)见表 7.3。

表　7.3

i	x_i	数值解 y_i	精确解 $(x_i+0.1)\mathrm{e}^{-2x_i}$	i	x_i	数值解 y_i	精确解 $(x_i+0.1)\mathrm{e}^{-2x_i}$
2	0.100 0	0.163 6	0.163 7	12	0.600 0	0.210 6	0.210 8
4	0.200 0	0.200 8	0.201 1	14	0.700 0	0.197 1	0.197 3
6	0.300 0	0.219 2	0.219 5	16	0.800 0	0.181 6	0.181 7
8	0.400 0	0.224 4	0.224 7	18	0.900 0	0.165 2	0.165 3
10	0.500 0	0.220 5	0.220 7	20	1.000 0	0.148 8	0.148 9

7.2.4　截断误差

定义(7.2)　对于数值方法,假定在前 i 个节点处的数值解等于其精确解,即 $y_i=y(x_i)$,则称

$$R_{i+1}=y_{i+1}-y(x_{i+1}) \tag{7.12}$$

为数值方法在节点 x_{i+1} 处的局部截断误差。它仅是从 i 到 $(i+1)$ 这一步计算方法引起的误差,在一定程度上反映了数值方法的精度。

定义(7.3)　若一种数值方法的局部截断误差为

$$R_{i+1}=o(h^{p+1}) \tag{7.13}$$

则称该数值方法具有 p 阶精度或是 p 阶方法。

这里只讨论欧拉方法的局部截断误差。为此将 $y(x_{i+1})$ 在节点 x_i 处用泰勒级数展开

$$y(x_{i+1})=y(x_i)+hf(x_i,y(x_i))+\frac{h^2}{2}y''(\xi_i)\quad (x_i\leqslant\xi\leqslant x_{i+1}) \tag{7.14}$$

从中减去欧拉公式(7.3),得到

$$R_{i+1}=-\frac{h^2}{2}y''(\xi_i)\quad (x_i\leqslant\xi\leqslant x_{i+1}) \tag{7.15}$$

可见,欧拉公式具有一阶精度,是一阶方法。还可以证明,梯形公式具有二阶精度,是二阶方法;预估-校正公式也具有二阶精度。

例 7.1　分别用欧拉方法和改进的欧拉方法求解初值问题

$$\begin{cases} y'=y-\dfrac{2x}{y} \\ y(0)=1 \end{cases}$$

在区间[0,1]上步长 $h=0.1$ 的数值解。

解　本问题的解析解是 $y=\sqrt{1+2x}$。用欧拉方法求解的计算公式是

$$\begin{cases} \bar{y}_{i+1} = y_i + 0.1\left(y_i - \dfrac{2x_i}{y_i}\right) \\ y_0 = 1 \end{cases} \quad (i=0,1,\cdots,9)$$

用改进的欧拉方法求解的计算公式是

$$\begin{cases} \bar{y}_{i+1} = y_i + 0.1\left(y_i - \dfrac{2x_i}{y_i}\right) \\ y_{i+1} = y_i + \dfrac{0.1}{2}\left[\left(y_i - \dfrac{2x_i}{y_i}\right) + \left(\bar{y}_{i+1} - \dfrac{2x_{i+1}}{\bar{y}_{i+1}}\right)\right] \end{cases} \quad (i=0,1,\cdots,9)$$

计算结果列入表 7.4 中。

表 7.4

x_i	y_i(欧拉方法)	y_i(改进的欧拉方法)	y_i(准确值)
0.1	1.100 000	1.095 909	1.095 445
0.2	1.191 818	1.184 097	1.183 216
0.3	1.277 438	1.266 201	1.264 911
0.4	1.358 213	1.343 360	1.341 641
0.5	1.435 133	1.416 402	1.414 214
0.6	1.508 966	1.485 956	1.483 240
0.7	1.580 338	1.552 515	1.549 193
0.8	1.649 783	1.616 475	1.612 451
0.9	1.717 779	1.678 168	1.673 320
1.0	1.784 778	1.737 867	1.732 051

可以看出,改进的欧拉方法比欧拉方法的计算精度要高。

7.3 龙格-库塔方法

7.3.1 龙格-库塔方法的基本思想

由微分中值定理可知,存在 $\theta \in (0,1)$,使得

$$\frac{y(x_{i+1}) - y(x_i)}{h} = y'(x_i + \theta h) = f(x_i + \theta h, y(x_i + \theta h)) \tag{7.16}$$

所以

$$y(x_{i+1}) = y(x_i) + hf(x_i + \theta h, y(x_i + \theta h)) \tag{7.17}$$

这里 $y'(x_i + \theta h) = f(x_i + \theta h, y(x_i + \theta h)) = \bar{k}$ 可以看作曲线 $y(x)$ 在区间 $[x_i, x_{i+1}]$ 上的平均斜率。由于 $y'(x_i + \theta h)$ 无法准确知道,故常用不同方法对其进行近似。

欧拉公式中用曲线在 x_i 处斜率作为平均斜率的近似值,具有一阶精度;梯形公式用曲线在 x_i 和 x_{i+1} 两点处斜率的算数平均值作为平均斜率的近似值,具有二阶精度。这就启发人

们，如果计算区间 $[x_i, x_{i+1}]$ 内多点处曲线的斜率，再把它们的某种线性组合作为曲线在 $[x_i, x_{i+1}]$ 内平均斜率的近似值，则有可能构造出精度更高的计算公式。这就是龙格-库塔(Runge-Kutta) 方法(亦简称为 R-K 方法) 的基本思想。

R-K 方法不采用导数值来构造平均斜率的近似计算公式，而是用不同点上函数值的适当线性组合来近似平均斜率，以此构造近似计算公式；再把近似计算公式的泰勒展开式与精确解的泰勒展开式进行比较，要求两展开式的前若干项相同，使近似计算公式达到一定精度。

具体构造时，先引进若干参数，如一般显式 R-K 方法的形式是

$$\left.\begin{aligned} &y_{i+1} = y_i + h\sum_{j=1}^{r}\omega_j k_j \\ &k_1 = f(x_i, y_i) \\ &k_j = f\left(x_i + \alpha_j h, y_i + h\sum_{l=1}^{j-1}\beta_{jl}k_l\right) \quad (j = 2, 3, \cdots, r) \end{aligned}\right\} \tag{7.18}$$

其中，ω_j，α_j 和 β_{jl} 是常数。选择它们的原则是，使上式右端在 (x_i, y_i) 处的泰勒展开式

$$y_{i+1} = y_i + \gamma_1 h + \frac{1}{2!}\gamma_2 h^2 + \frac{1}{3!}\gamma_3 h^3 + \cdots \tag{7.19}$$

与微分方程的精确解 $y(x)$ 在 x_i 处的泰勒展开式

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \frac{1}{2!}h^2 y''(x_i) + \cdots + \frac{1}{p!}h^p y^{(p)}(x_i) + o(h^{p+1}) \tag{7.20}$$

的前部尽可能多的项相同。如要求

$$\gamma_1 = f_i, \quad \gamma_2 = f'_i, \quad \gamma_3 = f''_i, \cdots\cdots, \quad \gamma_p = f_i^{(p-1)} \tag{7.21}$$

就得到 p 个方程，从中确定出 ω_j，α_j 和 β_{jl}，再代入 $k_1, k_2, \cdots\cdots, k_r$ 的表达式，就可以得到计算微分方程初值问题的数值计算公式

$$y_{i+1} = y_i + h\sum_{j=1}^{r}\omega_j k_j \tag{7.22}$$

这就是 p 阶 r 段的 R-K 方法计算公式。

定义(7.4)　若式(7.19) 与式(7.20) 的前 $p+1$ 项完全相同，即其局部截断误差达到 $o(h^{p+1})$，则称公式(7.18) 为 p 阶 r 段的 R-K 方法。

7.3.2　二阶龙格-库塔公式的推导

取 $p = r = 2$，即二阶龙格-库塔方法的计算公式是

$$\left.\begin{aligned} &y_{i+1} = y_i + h(\omega_1 k_1 + \omega_2 k_2) \\ &k_1 = f(x_i, y_i) \\ &k_2 = f(x_i + \alpha_2 h, y_i + \beta_{21}k_1 h) \end{aligned}\right\} \tag{7.23}$$

将上式中的 y_{i+1} 在 (x_i, y_i) 处展开，即

$$\begin{aligned} y_{i+1} = y_i + h(\omega_1 k_1 + \omega_2 k_2) = y_i + \omega_1 h f_i + \omega_2 h\left[f_i + h\left(\alpha_2\frac{\partial f_i}{\partial x} + \beta_{21}\frac{\partial f_i}{\partial y}f_i\right)\right] + o(h^3) = \\ y_i + (\omega_1 + \omega_2) f_i h + \omega_2\left(\alpha_2\frac{\partial f_i}{\partial x} + \beta_{21}\frac{\partial f_i}{\partial y}f_i\right)h^2 + o(h^3) \end{aligned}$$

式中，$f_i = f(x_i, y_i)$。上式与微分方程的精确解 $y = y(x)$ 在点 x_i 的展开式

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + y'(x_i)h + \frac{1}{2!}y''(x_i)h^2 + o(h^3) =$$

$$y_i + f_i h + \frac{1}{2}\left(\frac{\partial f_i}{\partial x} + \frac{\partial f_i}{\partial y}f_i\right)h^2 + o(h^3)$$

逐项比较，为了使

$$y(x_{i+1}) - y_{i+1} = o(h^3) \tag{7.24}$$

令 h 的一、二次项系数对应相等，有

$$\left.\begin{aligned} &\omega_1 + \omega_2 = 1 \\ &\omega_2 \alpha_2 = 1/2 \\ &\omega_2 \beta_{21} = 1/2 \end{aligned}\right\} \tag{7.25}$$

从中确定出 $\omega_1, \omega_2, \alpha_2$ 和 β_{21}，代入 R－K 计算公式(7.23)，即得到二阶两段的 R－K 方法计算公式。

方程组式(7.25)有无数组解，所以可以得到无穷多个二阶两段的 $R-K$ 方法计算公式。若取 $\omega_1 = 1/2, \omega_2 = 1/2, \alpha_2 = 1, \beta_{21} = 1$，就得到预估-校正公式(7.11)；若取 $\omega_1 = 0, \omega_2 = 1, \alpha_2 = 1/2, \beta_{21} = 1/2$，就得中点公式

$$\left.\begin{aligned} &y_{i+1} = y_i + hk_2 \\ &k_1 = f(x_i, y_i) \\ &k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) \end{aligned}\right\} \tag{7.26}$$

它也是常用的二阶公式。

7.3.3 三阶、四阶龙格-库塔公式举例

这里不作推导，给出几个比较常用的龙格-库塔公式。

(1) 三阶龙格-库塔公式(亦称库塔公式)

$$\left.\begin{aligned} &y_{i+1} = y_i + \frac{h}{6}(k_1 + 4k_2 + k_3) \\ &k_1 = f(x_i, y_i) \\ &k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) \\ &k_3 = f(x_i + h, y_i - hk_1 + 2hk_2) \end{aligned}\right\} \tag{7.27}$$

(2) 标准四阶龙格-库塔公式

$$\left.\begin{aligned} &y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ &k_1 = f(x_i, y_i) \\ &k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) \\ &k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right) \\ &k_4 = f(x_i + h, y_i + hk_3) \end{aligned}\right\} \tag{7.28}$$

(3) 吉尔(Gill)公式

$$\left.\begin{aligned}
&y_{i+1}=y_i+\frac{h}{6}\left[k_1+(2-\sqrt{2})k_2+(2+\sqrt{2})k_3+k_4\right]\\
&k_1=f(x_i,y_i)\\
&k_2=f\left(x_i+\frac{h}{2},y_i+\frac{h}{2}k_1\right)\\
&k_3=f\left(x_i+\frac{h}{2},y_i+\frac{\sqrt{2}-1}{2}hk_1+\frac{2-\sqrt{2}}{2}hk_2\right)\\
&k_4=f\left(x_i+h,y_i-\frac{\sqrt{2}}{2}hk_2+\frac{2+\sqrt{2}}{2}hk_3\right)
\end{aligned}\right\}\tag{7.29}$$

龙格-库塔方法的主要优点是计算精度较高，能满足通常的计算要求，程序也容易编制，而且可以从初始条件自动进行计算。其缺点是计算函数值的次数较多，计算工作量大，并且截断误差难以估计。在实际计算中，常常通过缩小步长来提高精度，而不是采用更高阶的计算公式。

例 7.2　取步长 $h=0.2$，从 $x=0$ 直到 $x=1$，用标准龙格-库塔方法求解初值问题

$$\begin{cases}y'=y-\dfrac{2x}{y}\\ y(0)=1\end{cases}$$

解　已知

$$f(x,y)=y-\frac{2x}{y},\quad y(0)=1,\quad h=0.2$$

求 $x=0.2,0.4,0.6,0.8,1.0$ 上的数值解。

将 $f(x,y)$ 和 $h=0.2$ 代人标准四阶龙格-库塔公式(7.28)，得出具体计算公式是

$$\begin{cases}
y_{i+1}=y_i+\dfrac{0.2}{6}(k_1+2k_2+2k_3+k_4)\\
k_1=y_i-\dfrac{2x_i}{y_i}\\
k_2=y_i+0.1k_1-\dfrac{2(x_i+0.1)}{y_i+0.1k_1}\\
k_3=y_i+0.1k_2-\dfrac{2(x_i+0.1)}{y_i+0.1k_2}\\
k_4=y_i+0.2k_3-\dfrac{2(x_i+0.2)}{y_i+0.2k_3}
\end{cases}\qquad(i=0,1,2,\cdots)$$

从 $i=0$ 开始，用上式的后四式计算一组 $k_m(m=1,2,3,4)$，代人上式的第一式，计算出 y_1；然后对 $i=1$ 进行计算，用上式的后四式计算一组 $k_m(m=1,2,3,4)$，代人上式的第一式，计算出 y_2；依此类推，可以求出方程精确解在所求节点处的数值计算结果(见表 7.5)。

表　7.5

i	x_i	y_i	i	x_i	y_i
0	0.0	1.000 00	3	0.6	1.483 28
1	0.2	1.183 23	4	0.8	1.612 51
2	0.4	1.341 67	5	1.0	1.732 14

程序(7.4) 用标准四阶龙格-库塔公式解微分方程初值问题的 MATLAB 程序。

程序任务：用标准四阶龙格-库塔公式(7.28)解初值问题式(7.1)。

```
function S = DeqRunKu4 (dfun, a, b, y0, Nm)
% 输入：dfun—— 微分方程中的函数 f(x,y)
%       a,b—— 求解区间两端的坐标
%       y0—— 解函数 y(x) 的初始值 y0
%       Nm—— 区间[a, b] 被等分的数目
% 输出：S—— S = [x' y'],其中 x 是数值解的节点坐标向量(1×(Nm+1)),y 是解函数在节点处的值
%            向量(1×(Nm+1))
h = (b - a)/Nm;              % 相邻两个节点之间的距离
x = a:h:b;                   % 节点坐标向量
y = zeros(1, Nm + 1);        % 初始化解函数值向量
y(1) = y0;                   % 赋初始值
for i = 1:Nm
    k1 = feval(dfun, x(i), y(i));                     % 按照式(7.28)第二式计算 k1
    k2 = feval(dfun, x(i) + h/2, y(i) + h * k1/2);    % 按照式(7.28)第三式计算 k2
    k3 = feval(dfun, x(i) + h/2, y(i) + h * k2/2);    % 按照式(7.28)第四式计算 k3
    k4 = feval(dfun, x(i) + h, y(i) + h * k3);        % 按照式(7.28)第五式计算 k4
    y(i+1) = y(i) + (k1 + 2 * k2 + 2 * k3 + k4) * h/6; % 按照式(7.28)第一式计算 y_{i+1}
end
S = [x' y'];          % 输出计算结果[x' y']
```

用程序(7.4)解初值问题 $\begin{cases} y' = e^{-2x} - 2y \\ y(0) = 0.1 \end{cases}$，取步长 $h = 0.3$，求解区间为[0, 3]。

在 MATLAB 命令窗口输入：

```
>> x=0:0.3:3;                                       % 建立节点向量
>> ya=exp(-2.*x)./10+x.*exp(-2.*x);                 % 计算精确解函数在节点处的值向量
>> dfun=inline('exp(-2*x)-2*y','x','y');            % 定义函数 f(x,y)=e^(-2x)-2y
>> S=DeqRunKu4 (dfun, 0, 3, 0.1, 10);               % 调用程序(7.4)求解(Nm=10)
>> [S ya']               % 输出：第一列是 x',第二列是数值解 y',第三列是精确解 ya'
```

输出结果见表 7.6。

表 7.6

i	x_i	数值解 y_i	精确解 $(x_i+0.1)e^{-2x_i}$	i	x_i	数值解 y_i	精确解 $(x_i+0.1)e^{-2x_i}$
1	0.300 0	0.219 1	0.219 5	6	1.800 0	0.051 9	0.051 9
2	0.600 0	0.210 4	0.210 8	7	2.100 0	0.033 0	0.033 0
3	0.900 0	0.165 0	0.165 3	8	2.400 0	0.020 6	0.020 6
4	1.200 0	0.117 8	0.117 9	9	2.700 0	0.012 7	0.012 6
5	1.500 0	0.079 6	0.079 7	10	3.000 0	0.007 7	0.007 7

7.3.4　步长的选择

步长越小，每步计算的截断误差越小。但在一定范围内求解时，步长的减小就意味着计算节点的增加，这不仅使计算量加大，同时可能导致舍入误差的积累加剧。为了保证计算精度，在实际计算中常常采用变步长技巧，即步长自动调整。

下面以四阶龙格-库塔方法为例，分析如何自动选择步长，以使计算精度满足要求。

从节点 x_i 开始，先以步长 h 计算出 $y(x_{i+1})$ 的近似值，记为 $y_{i+1}^{(h)}$ 。由于四阶公式的局部截断误差是 $o(h^5)$，故有

$$y(x_{i+1})-y_{i+1}^{(h)}\approx ch^5$$

当步长不大时，上式中的 c 可以看作常数。如果将步长减半，即取步长为 $h/2$，仍从节点 x_i 出发，经过两步计算得到 $y(x_{i+1})$ 的近似值，记为 $y_{i+1}^{(h/2)}$，考虑到每一步计算的截断误差均为 $c(h/2)^5$，于是就有

$$y(x_{i+1})-y_{i+1}^{(h/2)}\approx 2c\left(\frac{h}{2}\right)^5$$

以上两式相比可得

$$\frac{y(x_{i+1})-y_{i+1}^{(h/2)}}{y(x_{i+1})-y_{i+1}^{(h)}}\approx\frac{1}{16}\tag{7.30}$$

整理后是

$$y(x_{i+1})-y_{i+1}^{(h/2)}\approx\frac{1}{15}[y_{i+1}^{(h/2)}-y_{i+1}^{(h)}]\tag{7.31}$$

上式表明，若用 $y_{i+1}^{(h/2)}$ 作为 $y(x_{i+1})$ 的近似值，则误差约为先后两次计算结果之差的 1/15。

因此只要

$$|y_{i+1}^{(h/2)}-y_{i+1}^{(h)}|<\varepsilon\tag{7.32}$$

成立，就可将 $y_{i+1}^{(h/2)}$ 作为 $y(x_{i+1})$ 的近似值。若上式不成立，则将步长再次减半进行计算，直到上面的不等式成立为止，并取最后的 $y_{i+1}^{(h/2^k)}$ 作为计算结果。

以上方法就是计算过程中的步长自动选择方法，也称为变步长方法。

7.4　收敛性与稳定性

微分方程数值解法的收敛性和稳定性从不同角度反映了数值方法的可靠性，收敛性描述数值方法的合理程度，稳定性描述数值计算过程中误差的积累能否得到控制。只有收敛且稳定的方法，才能给出可信的计算结果。这里介绍单步法的收敛性和稳定性。

显式单步法的迭代公式能够统一表示为

$$y_{i+1}=y_i+h\varphi(x_i,y_i,h)\tag{7.33}$$

其中，$h=x_{i+1}-x_i$ 为步长，y_i 是方程精确解 $y(x)$ 在点 x_i 处的近似值，函数 $\varphi(x,y,h)$ 称为方法的增量函数。欧拉方法的增量函数 $\varphi(x,y,h)=f(x,y)$ 。

7.4.1　收敛性

微分方程初值问题的数值求解方法就是通过离散化将微分方程转化为差分方程。但问题

是，这种离散方法是否合适，即差分计算公式(7.33)的解在步长 $h \to 0$(同时 $i \to \infty$)时可否收敛到微分方程的精确解。

定义(7.5) 若数值方法对任意固定节点 $x_i = a + ih\ (i = 0, 1, 2, \cdots)$，当 $h \to 0$(且 $i \to \infty$)时，都有 $y_i \to y(x_i)$，则称该数值方法是收敛的。

数值方法的收敛性与方法本身的截断误差有关，而与数值计算时的舍入误差无关。

定理(7.2) 若

(1) 单步法迭代公式 $y_{i+1} = y_i + h\varphi(x_i, y_i, h)$ 中的增量函数 $\varphi(x, y, h)$ 在区域 $S = \{(x, y) \mid a \leqslant x \leqslant b, -\infty < y < +\infty, 0 < h < h_0\}$ 上连续，且关于 y 满足李普希兹条件：

$$|\varphi(x, y_1, h) - \varphi(x, y_2, h)| \leqslant L|y_1 - y_2| \quad (L > 0) \tag{7.34}$$

(2) 方法的局部截断误差 $R_i = o(h^{p+1})\ (i = 1, 2, \cdots)$；

(3) 初始值是精确的，即 $y(a) = y_0$。

则

(1) 方法的整体截断误差为 $e_i = o(h^p)$；

(2) 当 $p \geqslant 1$ 时，方法是收敛的。

例 7.3 对于试验方程(亦称模型方程)初值问题

$$\left.\begin{array}{l} y' = \lambda y \\ y(0) = y_0 \end{array} \quad (\lambda > 0)\right\} \tag{7.35}$$

分析欧拉方法的收敛性。

解 该问题的精确解是 $y(x) = y_0 e^{\lambda x}$。欧拉方法的迭代公式为

$$y_{i+1} = (1 + \lambda h)y_i$$

逐步迭代有

$$y_i = (1 + \lambda h)^i y_0 = y_0 (1 + \lambda h)^{\frac{1}{\lambda h}\lambda ih}$$

在本问题中，$x_0 = 0$，$x_i = ih$。而且当 $h \to 0$ 时 $(1 + \lambda h)^{\frac{1}{\lambda h}} \to e$，所以上式当 $h \to 0$ 时收敛到初值问题的精确解。

也可以根据定理(7.2)来证明本初值问题欧拉求解方法的收敛性。

7.4.2 稳定性

收敛性的分析是在认为数值计算完全精确的条件下进行的，而实际数值计算总有舍入误差，如果舍入误差随着迭代过程的进行不断积累放大，那么计算结果也会严重偏离问题的精确解，失去意义，这种数值方法就是不稳定的。

定义(7.6) 设数值方法计算 y_i 时，所得计算结果为 $\bar{y}_i$，由误差 $\delta_i = y_i - \bar{y}_i$ 引起以后各节点处计算结果的误差为 $\delta_m\ (m = i + 1, i + 2, \cdots)$，如果总有 $|\delta_m| \leqslant |\delta_i|$，则称该数值方法是绝对稳定的。

数值计算方法稳定性的分析相当复杂，它不仅与方法本身有关，而且与微分方程中的函数 $f(x, y)$ 及步长有关，所以研究一般微分方程数值求解方法的稳定性是很困难的。通常只对实验方程

$$\frac{dy}{dx} = \lambda y \tag{7.36}$$

(其中 λ 为常数,λ 为复数时,$\mathrm{Re}(\lambda)<0$) 进行讨论,即将数值方法用于解方程式(7.36),检查得到的差分方程是否数值稳定。应当注意,对实验方程绝对稳定的数值方法,对一般方程不一定也是绝对稳定的。

例 7.4　讨论欧拉方法对初值问题式(7.35) 的稳定性。

解　对于初值问题式(7.35),欧拉公式是

$$y_{i+1}=y_i+hf(x_i,y_i)=S(\lambda h)y_i$$

其中 $S(\lambda h)=1+\lambda h$。于是误差 δ_i 满足

$$|\delta_{i+1}|=|S(\lambda h)||\delta_i| \tag{7.37}$$

当 $|S(\lambda h)|<1$ 时,$|\delta_{i+1}|<|\delta_i|$。由此可得

$$|\delta_m|<|\delta_i| \quad (m=i+1,i+2,\cdots)$$

称区域 $\Omega=\{\mu\,||S(\lambda h)|<1\}$ 为欧拉公式的绝对稳定区域。因此,适当选择步长,使 $|S(\lambda h)|<1$,就能保证数值方法的稳定性。

7.5　常微分方程组与高阶常微分方程的求解

前面介绍的常微分方程数值求解方法的思路同样可以用于一阶常微分方程组和高阶方程的求解。

7.5.1　常微分方程组的求解

设有一阶微分方程组的初值问题

$$\left.\begin{aligned}\frac{\mathrm{d}y}{\mathrm{d}x}&=f_1(x,y,z), \quad y(x_0)=y_0\\ \frac{\mathrm{d}z}{\mathrm{d}x}&=f_2(x,y,z), \quad z(x_0)=z_0\end{aligned}\right\} \tag{7.38}$$

方程组的解是一组可微函数 $y(x)$ 和 $z(x)$,当它们代入方程组式(7.38) 时就有

$$\left.\begin{aligned}y'(x)&=f_1(x,y(x),z(x)), \quad y(x_0)=y_0\\ z'(x)&=f_2(x,y(x),z(x)), \quad z(x_0)=z_0\end{aligned}\right\} \tag{7.39}$$

为了建立方程组的数值求解方法,考虑微分

$$\left.\begin{aligned}\mathrm{d}y&=f_1(x,y,z)\,\mathrm{d}x\\ \mathrm{d}z&=f_2(x,y,z)\,\mathrm{d}x\end{aligned}\right\} \tag{7.40}$$

令 $\mathrm{d}x=x_{k+1}-x_k$,$\mathrm{d}y=y_{k+1}-y_k$,$\mathrm{d}z=z_{k+1}-z_k$,得

$$\left.\begin{aligned}y_{k+1}-y_k&\approx f_1(x_k,y_k,z_k)(x_{k+1}-x_k)\\ z_{k+1}-z_k&\approx f_2(x_k,y_k,z_k)(x_{k+1}-x_k)\end{aligned}\right\} \tag{7.41}$$

将区间分为若干个子区间,区间的宽度为 h、边界为 $x_{k+1}=x_k+h(k=0,1,2,\cdots)$,从式(7.41) 就可以得到欧拉方法的递推公式

$$\left.\begin{aligned}y_{k+1}&=y_k+hf_1(x_k,y_k,z_k), \quad y_0=y(x_0)\\ z_{k+1}&=z_k+hf_2(x_k,y_k,z_k), \quad z_0=z(x_0)\end{aligned}\quad (k=0,1,2,\cdots)\right\} \tag{7.42}$$

改进的欧拉公式是

$$\left.\begin{aligned}
&y_{k+1}^{(0)}=y_k+hf_1(x_k,y_k,z_k)\\
&z_{k+1}^{(0)}=z_k+hf_2(x_k,y_k,z_k)\\
&y_{k+1}^{(n+1)}=y_k+\frac{h}{2}[f_1(x_k,y_k,z_k)+f_1(x_k,y_{k+1}^{(n)},z_{k+1}^{(0)})]\qquad(k,n=0,1,2,\cdots)\\
&z_{k+1}^{(n+1)}=z_k+\frac{h}{2}[f_2(x_k,y_k,z_k)+f_2(x_k,y_{k+1}^{(0)},z_{k+1}^{(n)})]
\end{aligned}\right\}\tag{7.43}$$

当连续两次迭代结果满足精度要求,即

$$|y_{k+1}^{(n+1)}-y_{k+1}^{(n)}|<\varepsilon,\quad |z_{k+1}^{(n+1)}-z_{k+1}^{(n)}|<\varepsilon$$

时,取 $y_{k+1}=y_{k+1}^{(n+1)}$,$z_{k+1}=z_{k+1}^{(n+1)}$,转入下一步计算。

四阶龙格-库塔公式是

$$\left.\begin{aligned}
&y_{k+1}=y_k+\frac{h}{6}(l_1+2l_2+2l_3+l_4)\\
&z_{k+1}=z_k+\frac{h}{6}(m_1+2m_2+2m_3+m_4)\\
&l_1=f_1(x_k,y_k,z_k)\\
&m_1=f_2(x_k,y_k,z_k)\\
&l_2=f_1\left(x_k+\frac{h}{2},y_k+\frac{hl_1}{2},z_k+\frac{hm_1}{2}\right)\\
&m_2=f_2\left(x_k+\frac{h}{2},y_k+\frac{hl_1}{2},z_k+\frac{hm_1}{2}\right)\qquad(k=0,1,2,\cdots)\\
&l_3=f_1\left(x_k+\frac{h}{2},y_k+\frac{hl_2}{2},z_k+\frac{hm_2}{2}\right)\\
&m_3=f_2\left(x_k+\frac{h}{2},y_k+\frac{hl_2}{2},z_k+\frac{hm_2}{2}\right)\\
&l_4=f_1(x_k+h,y_k+hl_3,z_k+hm_3)\\
&m_4=f_2(x_k+h,y_k+hl_3,z_k+hm_3)
\end{aligned}\right\}\tag{7.44}$$

程序(7.5) 用四阶龙格-库塔公式解微分方程组初值问题的 MATLAB 程序。

程序任务:用四阶龙格-库塔公式(7.44)解微分方程组初值问题式(7.38)。

```
function S = DeqsRunKu (yfun, zfun, a, b, y0, z0, Nm)
% 输入:yfun—— 微分方程组中的函数 f1(x,y,z)
%      zfun—— 微分方程组中的函数 f2(x,y,z)
%      a, b—— 求解区间两端的坐标
%      y0—— 解函数 y(x) 的初始值 y0
%      z0—— 解函数 z(x) 的初始值 z0
%      Nm—— 区间[a, b] 被等分的数目
% 输出:S—— S = [x' y' z'],其中 x 是数值解的节点坐标向量(1 × (Nm + 1)),y 与 z 分别是解函数
%          y(x) 与 z(x) 在节点处的值向量(1 × (Nm + 1))
h = (b - a)/Nm;          % 相邻两个节点之间的距离
x = a:h:b;               % 节点坐标向量
y = zeros(1, Nm + 1);    % 初始化解函数值向量 y
z = zeros(1, Nm + 1);    % 初始化解函数值向量 z
y(1) = y0; z(1) = z0     % 赋初始值
```

```
for i = 1:Nm
    l1 = feval(yfun, x(i), y(i), z(i));                              % 按照式(7.44)第三式计算 l1
    m1 = feval(zfun, x(i), y(i), z(i));                              % 按照式(7.44)第四式计算 m1
    l2 = feval(yfun, x(i) + h/2, y(i) + h * l1/2, z(i) + h * m1/2);  % 按照式(7.44)第五式计算 l2
    m2 = feval(zfun, x(i) + h/2, y(i) + h * l1/2, z(i) + h * m1/2);  % 按照式(7.44)第六式计算 m2
    l3 = feval(yfun, x(i) + h/2, y(i) + h * l2/2, z(i) + h * m2/2);  % 按照式(7.44)第七式计算 l3
    m3 = feval(zfun, x(i) + h/2, y(i) + h * l2/2, z(i) + h * m2/2);  % 按照式(7.44)第八式计算 m3
    l4 = feval(yfun, x(i) + h, y(i) + h * l3, z(i) + h * m3);        % 按照式(7.44)第九式计算 l4
    m4 = feval(zfun, x(i) + h, y(i) + h * l3, z(i) + h * m3);        % 按照式(7.44)第十式计算 m4
    y(i + 1) = y(i) + (l1 + 2 * l2 + 2 * l3 + l4) * h/6;            % 按照式(7.44)第一式计算 yi+1
    z(i + 1) = z(i) + (m1 + 2 * m2 + 2 * m3 + m4) * h/6;            % 按照式(7.44)第二式计算 zi+1
end
S = [x' y' z'];  % 输出计算结果[x' y' z']
```

例 7.5　用程序(7.5)解初值问题

$$\begin{cases} \dfrac{dx}{dt} = x + 2y, & x(0) = 6 \\ \dfrac{dy}{dt} = 3x + 2y, & y(0) = 4 \end{cases}$$

计算区间为[0, 0.2]。

在 MATLAB 命令窗口输入：

```
>> yfun=inline('y + 2 * z','x','y','z');              % 定义函数 f1(x,y,z)=y+2z
>> zfun=inline('3 * y+2 * z','x','y','z');            % 定义函数 f2(x,y,z)=3y+2z
>> S=DeqsRunKu (yfun, zfun, 0, 2, 6, 4, 10);          % 调用程序(7.5)求解(Nm=10)
>> x=0:0.2:2;                                         % 输入节点向量[x0,x1,x2,…,xn]
>> y=4 * exp(4. * x) + 2. * exp(- x);                 % 计算精确解向量[y0,y1,y2,…,yn]
>> z=6 * exp(4. * x) - 2. * exp(- x);                 % 计算精确解向量[z0,z1,z2,…,zn]
>> [S y' z']          % 输出结果:第一列是x',第二列是数值解y',第三列是数值解z',
                        第四列是精确解[y(xi)]',第五列是精确解[z(xi)]'
```

输出结果见表 7.7。

表　7.7

i	t_i	数值解 x_i	精确解 $x(t_i) = 4e^{4t_i} + 2e^{-t_i}$	数值解 y_i	精确解 $y(t_i) = 6e^{4t_i} - 2e^{-t_i}$
1	0.2	11	11	12	12
2	0.4	21	21	28	28
3	0.6	45	45	65	65
4	0.8	98	99	145	146
5	1.0	218	219	325	327
6	1.2	483	487	722	728
7	1.4	1 072	1 082	1 606	1 622
8	1.6	2 381	2 408	3 570	3 611
9	1.8	5 290	5 358	7 935	8 036
10	2.0	11 757	11 924	17 635	17 885

7.5.2 高阶常微分方程的求解

二阶常微分方程的初值问题是

$$y''(x)=f(x,y(x),y'(x)),\quad y(x_0)=y_0,\quad y'(x_0)=z_0 \tag{7.45}$$

若令 $z(x)=y'(x)$，则初值问题式(7.45)转化为

$$\left.\begin{aligned} &z'(x)=f(x,y(x),z(x)), && z(x_0)=z_0 \\ &y'(x)=z(x), && y(x_0)=y_0 \end{aligned}\right\} \tag{7.46}$$

成为一阶常微分方程组。这时四阶龙格-库塔公式(7.44)成为

$$\begin{cases} z_{k+1}=z_k+\dfrac{h}{6}(l_1+2l_2+2l_3+l_4) \\ y_{k+1}=y_k+\dfrac{h}{6}(m_1+2m_2+2m_3+m_4) \end{cases} \quad (k=0,1,2,\cdots)$$

其中

$$\begin{cases} l_1=f(x_k,y_k,z_k) \\ l_2=f\left(x_k+\dfrac{h}{2},y_k+\dfrac{hm_1}{2},z_k+\dfrac{hl_1}{2}\right) \\ l_3=f\left(x_k+\dfrac{h}{2},y_k+\dfrac{hm_2}{2},z_k+\dfrac{hl_2}{2}\right) \\ l_4=f(x_k+h,y_k+hm_3,z_k+hl_3) \\ m_1=z_k \\ m_2=z_k+\dfrac{hl_1}{2} \\ m_3=z_k+\dfrac{hl_2}{2} \\ m_4=z_k+hl_3 \end{cases} \quad (k=0,1,2,\cdots)$$

从以上两式中消去 m_1,m_2,m_3 和 m_4，则四阶龙格-库塔公式可以进一步简化为

$$\left.\begin{aligned} &z_{k+1}=z_k+\frac{h}{6}(l_1+2l_2+2l_3+l_4) \\ &y_{k+1}=y_k+hz_k+\frac{h^2}{6}(l_1+l_2+l_3) \end{aligned}\quad (k=0,1,2,\cdots)\right\} \tag{7.47}$$

式中

$$\left.\begin{aligned} &l_1=f(x_k,y_k,z_k) \\ &l_2=f\left(x_k+\frac{h}{2},y_k+\frac{hz_k}{2},z_k+\frac{hl_1}{2}\right) \\ &l_3=f\left(x_k+\frac{h}{2},y_k+\frac{hz_k}{2}+\frac{h^2l_1}{4},z_k+\frac{hl_2}{2}\right) \\ &l_4=f\left(x_k+h,y_k+hz_k+\frac{h^2l_2}{2},z_k+hl_3\right) \end{aligned}\quad (k=0,1,2,\cdots)\right\} \tag{7.48}$$

程序(7.6) 用四阶龙格-库塔公式解二阶微分方程初值问题的MATLAB程序。

程序任务：用四阶龙格-库塔公式(7.47)解二阶常微分方程的初值问题式(7.45)。

```
function S = SDeqRunKu (dfun, a, b, y0, z0, Nm)
% 输入:dfun—— 微分方程中的函数 f(x,y(x),z(x))
```

```
%      a, b—— 求解区间两端的坐标
%      z0——z(x) = y′(x) 的初始值 z0 = y′(x0)
%      y0—— 解函数 y(x) 的初始值 y0 = y(x0)
%      Nm—— 区间[a, b]被等分的数目
% 输出:S—— S = [x′ y′],其中 x 是数值解的节点坐标向量(1×(Nm+1)),y 是解函数 y(x) 在节点处
              的近似值向量(1×(Nm+1))
h = (b - a)/Nm;                % 相邻两个节点之间的距离
x = a:h:b;                     % 节点坐标向量
y = zeros(1, Nm + 1);          % 初始化解函数值向量 y
z = zeros(1, Nm + 1);          % 初始化向量 z
y(1) = y0; z(1) = z0           % 赋初始值
for i = 1:Nm
    l1 = feval(dfun, x(i), y(i), z(i));                          % 按照式(7.48)第一式计算 l1
    l2 = feval(dfun, x(i) + h/2, y(i) + h * z(i)/2, z(i) + h * l1/2);  % 按照式(7.48)第二式计算 l2
    l3 = feval(dfun, x(i) + h/2, y(i) + h * z(i)/2 + h^2 * l1/4, z(i) + h * l2/2);
                                                                   % 按照式(7.48)第三式计算 l3
    l4 = feval(dfun, x(i) + h, y(i) + h * z(i) + h^2 * l2/2, z(i) + h * l3);  % 按照式(7.48)第四式计算 l4
    z(i + 1) = z(i) + (l1 + 2 * l2 + 2 * l3 + l4) * h/6;          % 按照式(7.47)第一式计算 z_{i+1}
    y(i + 1) = y(i) + h * z(i) + (l1 + l2 + l3) * h^2/6;          % 按照式(7.44)第二式计算 y_{i+1}
end
S = [x′ y′];  % 输出计算结果[x′ y′]
```

例 7.6　用程序(7.6)计算区间[0,2]内二阶微分方程

$$x''(t)+4x'(t)+5x(t)=0,\quad x(0)=3,\quad x'(0)=-5$$

的数值解,将区间分成 10 个子区间。

在 MATLAB 命令窗口输入:

```
>> dfun = inline('- 5 * y - 4 * z','x','y','z');     % 定义函数 f(x,y,z) = -5y - 4z
>> S = SDeqRunKu (dfun, 0, 2, 3, - 5, 10);           % 调用程序(7.6)求解(Nm = 10)
>> x = 0:0.2:2;                                      % 输入节点向量[x0,x1,x2,…,xn]
>> ya = 3. * exp(- 2. * x). * cos(x) + exp(- 2. * x). * sin(x);
                                                     % 计算精确解 y(x) = e^{-2x}[3cosx + sinx]
>> [S ya']   % 输出结果:第一列是 x′,第二列是数值解 y′,第三列是精确解[y(x_i)]
```

输出结果见表 7.8。

表　7.8

i	x_i	数值解 y_i	精确解 $y(x_i)$	i	x_i	数值解 y_i	精确解 $y(x_i)$
1	0.200 0	2.103 7	2.104 0	6	1.200 0	0.183 2	0.183 2
2	0.400 0	1.416 1	1.416 6	7	1.400 0	0.091 0	0.090 9
3	0.600 0	0.915 5	0.915 8	8	1.600 0	0.037 3	0.037 2
4	0.800 0	0.566 6	0.566 8	9	1.800 0	0.008 1	0.008 0
5	1.000 0	0.333 1	0.333 2	10	2.000 0	−0.00 61	−0.00 62

7.6 物理学中的应用举例——单摆运动规律分析

如图7.2所示，单摆的摆线长度为l，小球质量为m，摆线相对于铅垂方向的角位移为θ，单摆受到的阻力与小球运动速度成正比，阻力系数为γ，在周期性驱动力$F\cos(\omega_d t)$作用下，单摆的运动方程是

$$ml\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}+\gamma l\frac{\mathrm{d}\theta}{\mathrm{d}t}+mg\sin\theta=F\cos(\omega_d t) \tag{7.49}$$

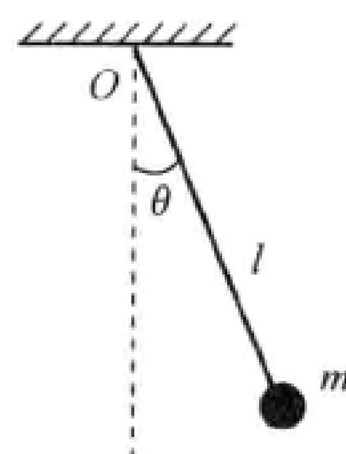


图 7.2

单摆的固有圆频率$\omega_0=\sqrt{g/l}$。引入无量纲量

$$\omega=\frac{\omega_d}{\omega_0},\quad \tau=\omega_0 t,\quad \beta=\frac{\gamma}{2m\omega_0},\quad f=\frac{F}{mg} \tag{7.50}$$

方程式(7.49)就可以无量纲化为

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\tau^2}+2\beta\frac{\mathrm{d}\theta}{\mathrm{d}\tau}+\sin\theta=f\cos(\omega\tau) \tag{7.51}$$

对式(7.49)各项乘以$\mathrm{d}\theta/mg$，有

$$\frac{1}{2}\mathrm{d}(\omega'^2)+2\beta\omega'^2\mathrm{d}\tau+\sin\theta\mathrm{d}\theta=f\omega'\cos(\omega\tau)\mathrm{d}\tau \tag{7.52}$$

式中，$\omega'=\mathrm{d}\theta/\mathrm{d}\tau$。设初始时刻，单摆的运动状态是$\theta_0=0$，$\omega'=\omega'_0$，对上式从初始时刻到某一状态进行积分，有

$$E=E_0-2\beta\int_0^\tau\omega'^2\mathrm{d}\tau+f\int_0^\tau\omega'\cos(\omega\tau)\mathrm{d}\tau \tag{7.53}$$

式中

$$E=\frac{1}{2}\omega'^2-\cos\theta,\quad E_0=\frac{1}{2}\omega'^2_0-1 \tag{7.54}$$

分别是无量纲化的机械能和无量纲化的初始时刻机械能(取系统的重力势能零参考面为过O点的水平面)。式(7.53)右端的第二项是无量纲化的阻力的功，第三项是无量纲化的驱动力的功。可以发现，阻力的功总是使系统的机械能减小，驱动力的功可能使系统的机械能增大，也可能使系统的机械能减小，这取决于ω'和$\cos(\omega\tau)$的符号能否保持基本相同。

下面分几种情况，应用数值方法解微分方程，对单摆运动规律进行分析。

7.6.1 无阻尼无驱动单摆的运动

对于不受阻力也无驱动力作用的单摆，$\beta=0$，$f=0$，方程式(7.51)和式(7.53)分别简化为

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\tau^2}+\sin\theta=0 \tag{7.55}$$

$$E=E_0=\frac{1}{2}\left(\frac{\mathrm{d}\theta}{\mathrm{d}\tau}\right)^2-\cos\theta\geqslant-1 \tag{7.56}$$

即机械能守恒。从式(7.56)可以导出

$$\left(\frac{\mathrm{d}\theta}{\mathrm{d}\tau}\right)^2=2(E_0+\cos\theta) \tag{7.57}$$

由于$(\mathrm{d}\theta/\mathrm{d}\tau)^2\geqslant 0$，从公式(7.57)能够得到以下结论：

(1) 若$E_0=-1$，则$\theta=0$，系统动能只能等于零，即单摆停止不动。

(2) 若 $-1 < E_0 < 1$，则 $-\pi < -\arccos(-E_0) \leqslant \theta \leqslant \arccos(-E_0) < \pi$，单摆的角位移在 $[-\arccos(-E_0), \arccos(-E_0)]$ 范围内变化，单摆作周期性振动。

(3) 若 $E_0 = 1$，则 $-\pi \leqslant \theta \leqslant \pi$，单摆的角位移可以在 $[-\pi, \pi]$ 范围内变化。当 $\theta = \pm\pi$ 时 $(\mathrm{d}\theta/\mathrm{d}\tau) = 0$，即单摆停止在此位置，不再运动。

(4) 若 $E_0 > 1$，则 $\mathrm{d}\theta/\mathrm{d}\tau = \pm\sqrt{2(E_0 + \cos\theta)} \neq 0$，而且在 θ 单方向变化的过程中，$\mathrm{d}\theta/\mathrm{d}\tau$ 的符号保持不变，说明单摆绕固定方向转动，作圆周运动。

图 7.3 给出了角位移 θ 与 τ 的关系曲线和角速度 ω' 与角位移 θ 的关系曲线。ω'-θ 平面称为相平面，在相平面上研究单摆运动的方法称为相图研究法。相平面上的一个点表示系统在某一时刻的状态(角位移与角速度)，称为相点，而相点连续变化形成的轨迹称为相轨线，它描述了系统的运动过程。图中的 7 条曲线分别表示初始状态为 $\theta_0 = 0$ 及 $\omega'_0 = \sqrt{i}\,(i = 0, 1, \cdots, 6)$ (对应的 $E_0 = -1, -1/2, 0, 1/2, 1, 3/2, 2$) 时，单摆的运动过程。曲线 1 的 $E_0 = -1$，单摆静止不动，θ-τ 曲线是一条水平直线($\theta = 0$)，ω'-θ 曲线是一个点($\theta = 0, \omega' = 0$)；曲线 2～4 满足 $-1 < E_0 < 1$，单摆作周期性振动，θ-τ 曲线是周期性变化的曲线，ω'-θ 曲线是闭合曲线(由于 $\omega'_0 > 0$，相点顺时针旋转)。单摆的机械能越大，闭合曲线包围的面积越大，单摆振动周期也越长；曲线 5 的 $\omega'_0 = 2, E_0 = 1$，小球摆到最高点时就停止不动，因此 θ-τ 曲线上升后成为水平直线($\theta = \pi$)，ω'-θ 曲线单调下降并终止在点$(\pi, 0)$；曲线 6 和曲线 7 满足 $E_0 > 1$，θ-τ 平面上的曲线单调增加，说明单摆作非周期性运动，相平面上对应的曲线是周期性振荡曲线，说明 ω' 是 θ 的周期函数，实际上小球在作圆周运动。总之，相平面上的四种曲线(点、闭合曲线、单调非闭合曲线及周期性振荡曲线)反映了无阻尼无驱动单摆的四种运动形式，运动形式与单摆的机械能大小有关。另外，根据对称性，可以得到相平面上其他三部分的相轨线。

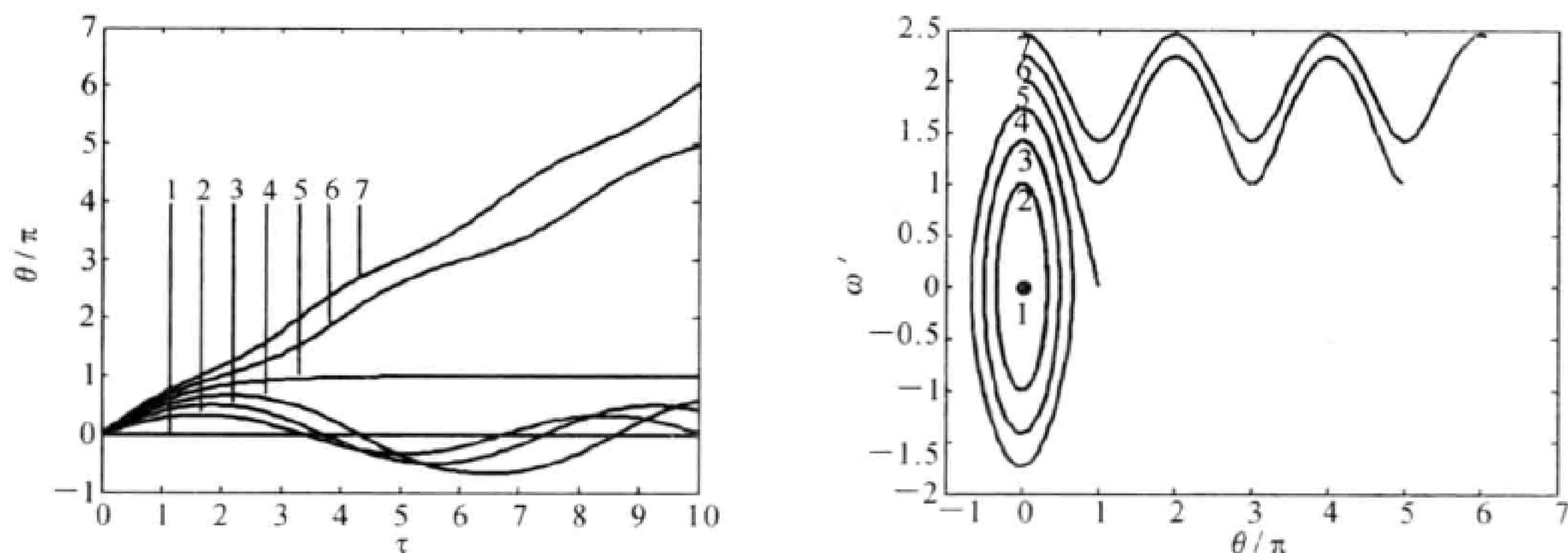


图　7.3

7.6.2　有阻尼无驱动单摆的运动

对于受阻力而无驱动力的单摆，$f = 0$，方程式(7.51)和式(7.53)分别简化为

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\tau^2} + 2\beta\frac{\mathrm{d}\theta}{\mathrm{d}\tau} + \sin\theta = 0 \tag{7.58}$$

$$E = E_0 - 2\beta\int_0^\tau \omega'^2\,\mathrm{d}\tau \geqslant -1 \tag{7.59}$$

式(7.59)说明,单摆的机械能随着时间的增加而减小,单摆最终会停止运动。

图7.4给出了$\beta=0.15$,初始状态为$\theta_0=0$与$\omega'_0=\sqrt{2i}$ ($i=0,1,\cdots,13$)(对应的$E_0=-1,0,\cdots,12$)时,单摆运动的$\theta-\tau$曲线和$\omega'-\theta$曲线,从下往上曲线分别与14种初始状态对应。图中显示,曲线分成了三组,在$\theta-\tau$平面上每一组曲线有共同的渐近线,在相平面上,每一组曲线终止于同一点。在$\theta-\tau$平面上,相邻渐近线对应的角位移相差2π,说明单摆绕中心转动的圈数相差1。在$\omega'-\theta$平面上,相邻终止点对应的角位移也相差2π,对应的角速度ω'都是零。说明,由于受到阻力的作用,初始时刻具有不同机械能的单摆,在绕悬挂点转动不同的圈数后,作减幅振动,最终归于静止。相平面上每组相轨线的共同终止点称为吸引子,似乎是它把相平面上的相点吸引了过来。

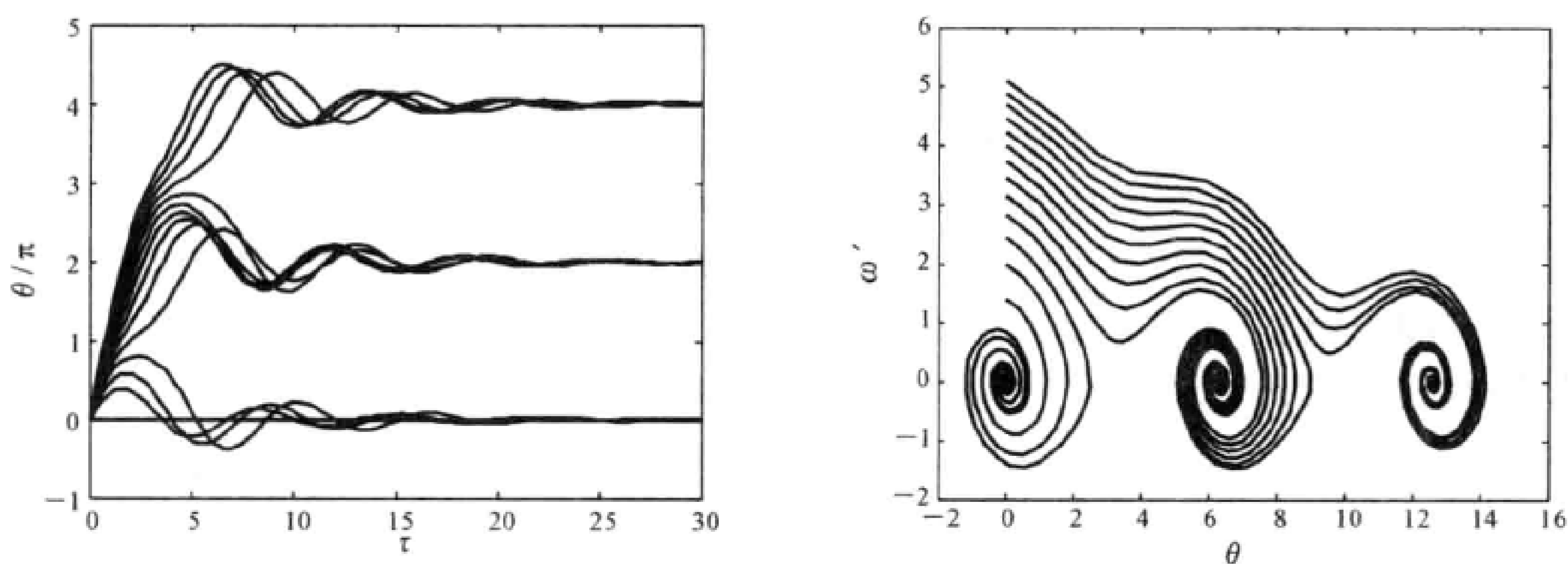


图 7.4

7.6.3 有阻尼有驱动单摆的运动

图7.5给出了$\beta=0.20$,$f=0.50$,初始状态为$\theta_0=0$与$\omega'_0=\sqrt{2i}$ ($i=0,1,\cdots,5$)(对应的$E_0=-1,0,\cdots,4$),并且驱动力圆频率ω不同时,单摆运动的$\theta-\tau$曲线和$\omega'-\theta$曲线。

从图7.5中可以看出,由于受到阻力和驱动力的共同作用,无论单摆的初始机械能如何,单摆在绕悬挂点转动几圈并经过减幅振动的过渡后,都会归于以驱动力周期为周期的振动。驱动力的频率和幅度不同,这种过渡的快慢不同,当驱动力的频率等于单摆固有频率($\omega=1$)时,过渡时间最短。在相平面上,相轨线最终形成闭合曲线。

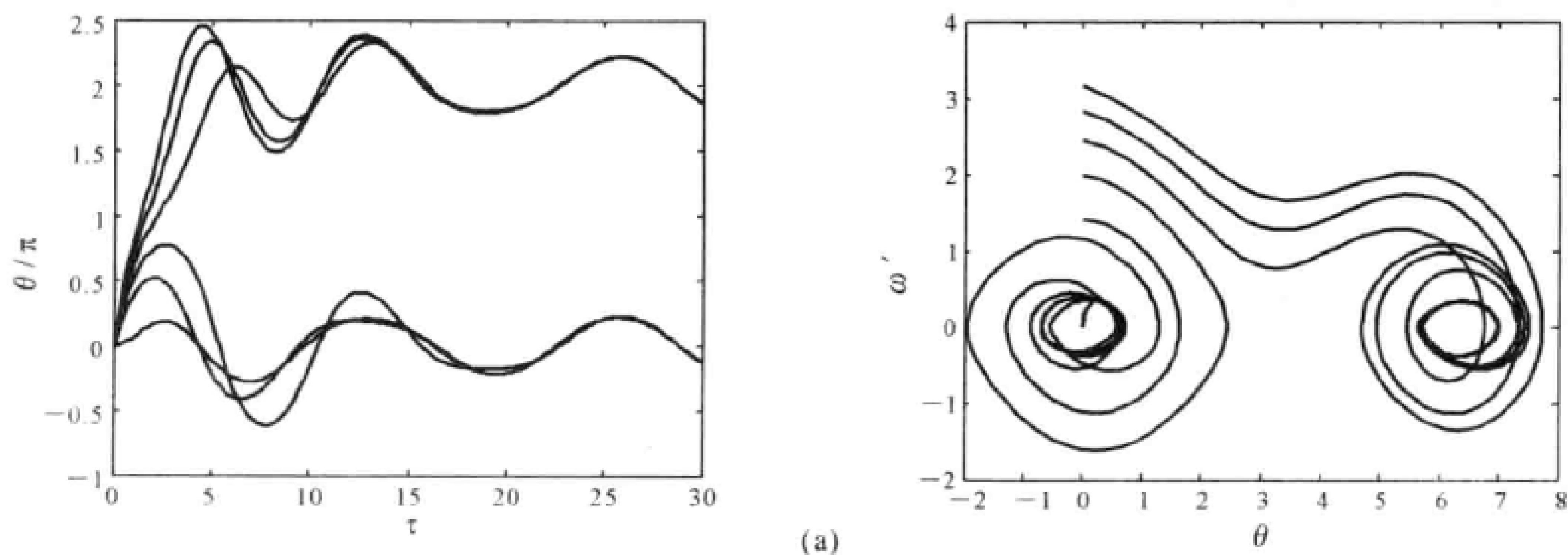
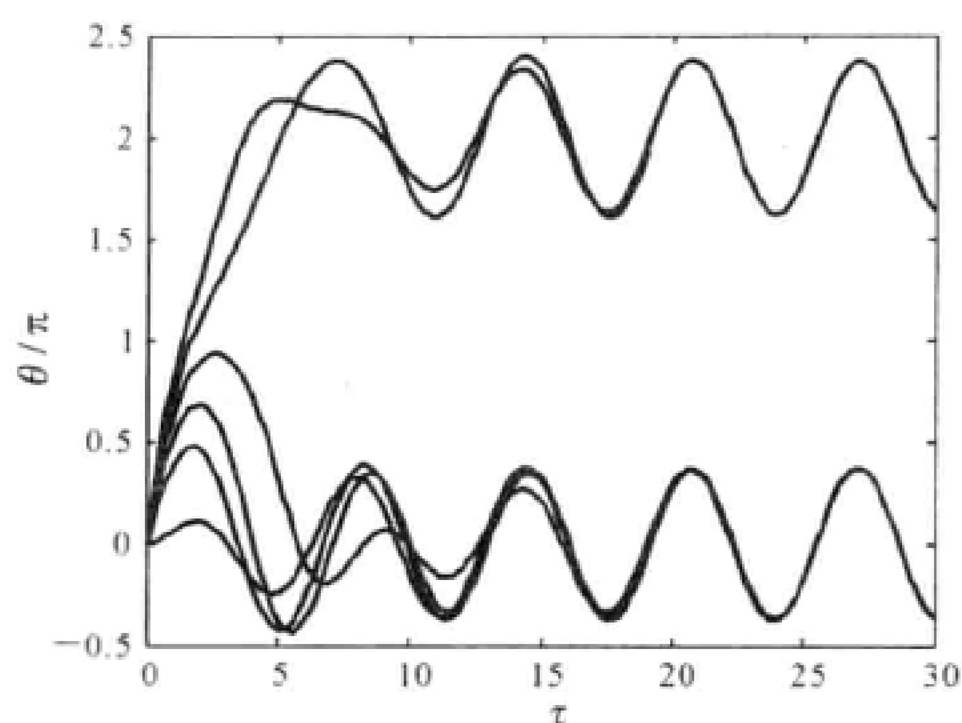
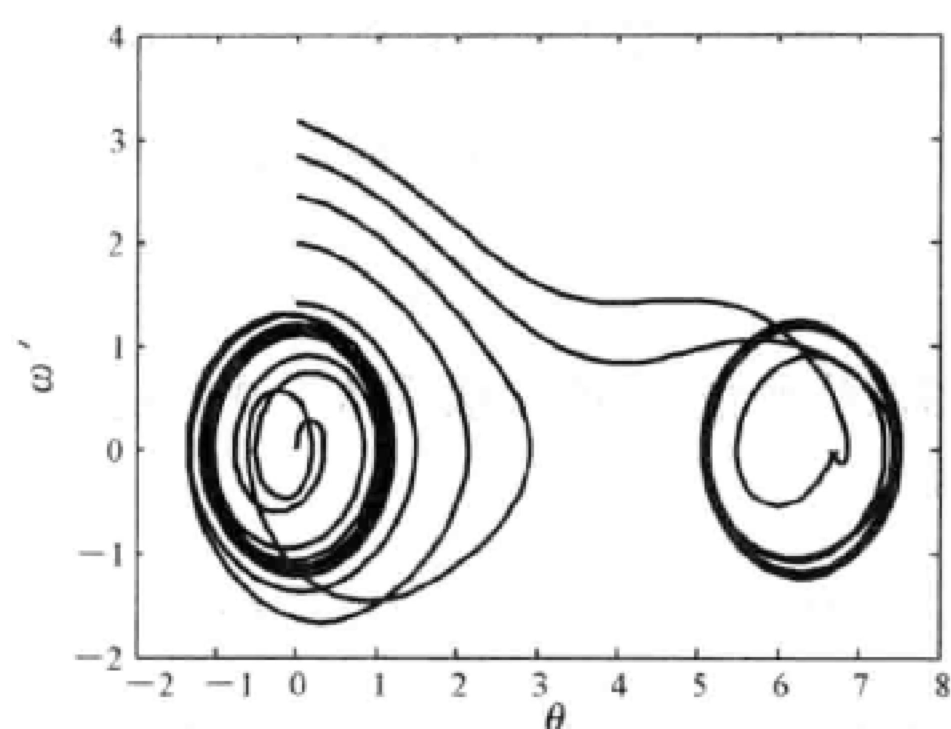


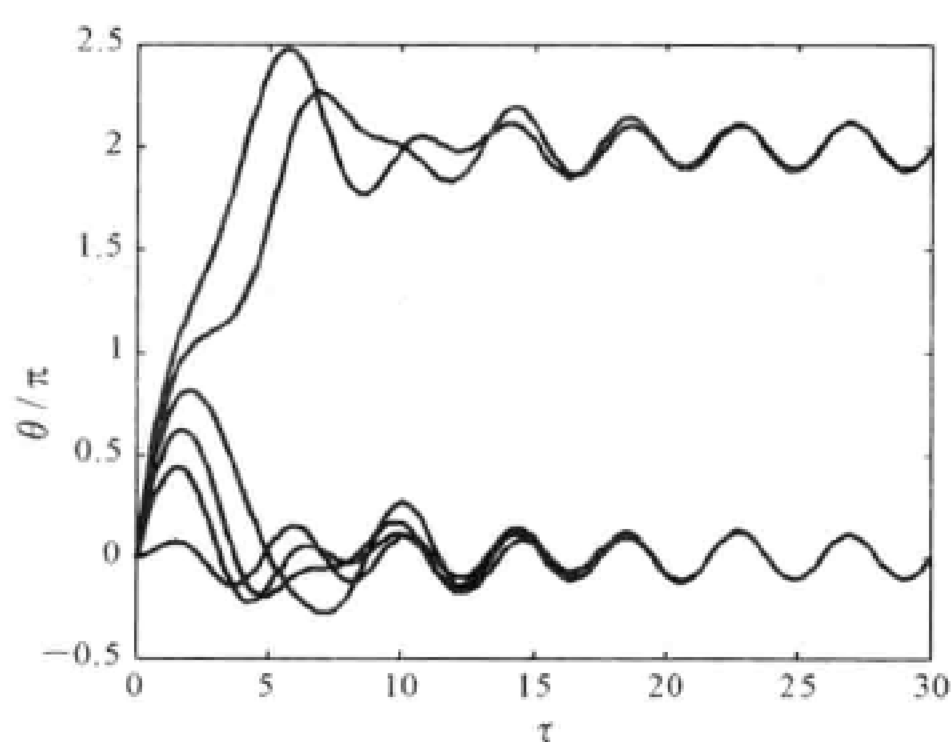
图 7.5

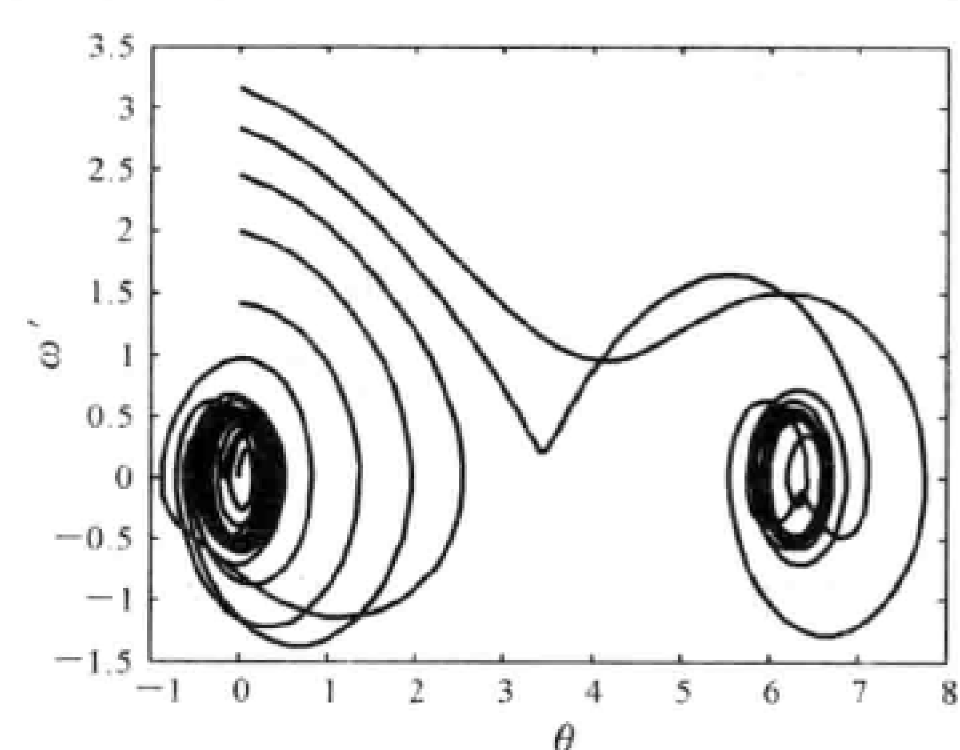
(a)$\omega=0.5$





(b)





(c)

续图　7.5

(b)$\omega = 1.0$；(c)$\omega = 1.5$

习　　题

1. 用欧拉方法求解下列初值问题：

(1) $\begin{cases} y' = -y + x + 1 \\ y(0) = 1 \end{cases}$ $(0 \leqslant x \leqslant 1, h = 0.1)$；　(2) $\begin{cases} y' = |\sin x| \\ y(0) = 1 \end{cases}$ $(0 \leqslant x \leqslant 2\pi, h = 0.1)$

2. 用改进的欧拉方法解初值问题 $\begin{cases} y' + y + xy^2 = 0 \\ y(0) = 1 \end{cases}$ $(0 \leqslant x \leqslant 1, h = 0.2)$，并将数值解与精确解 $y(x) = 1/(2e^x - x - 1)$ 进行比较。

3. 分别用龙格-库塔法的中点公式和标准四阶龙格-库塔公式解初值问题 $\begin{cases} y' = x + y \\ y(0) = 1 \end{cases}$ $(0 \leqslant x \leqslant 0.6, h = 0.2)$，并与精确解 $y(x) = 2e^x - x - 1$ 进行比较。

4. 在区间[0，8]内用 $h=0.1$ 求解微分方程组 $\begin{cases} x' = x - xy, & x(0) = 4 \\ y' = -y + xy, & y(0) = 1 \end{cases}$。该微分方程组的解描述的点 (x, y) 形成闭合曲线。

5. 用四阶龙格-库塔方法求解初值问题 $\begin{cases} y'' - 2y^3 = 0 \\ y(1) = y'(1) = -1 \end{cases}$ $(x \in (1, 1.5))$。

6. 用欧拉方法计算积分$\int_0^x e^{t^2}\,dt$在点$x=0.5,1,1.5,2$处的值。

7. 通过求解一个合适的初值问题，制作如下积分的函数表：

$$f(x)=\frac{1}{2}+\frac{1}{\sqrt{2\pi}}\int_0^x \exp\left(-\frac{t^2}{2}\right)dt,\quad (0\leqslant x\leqslant 3)$$

利用四阶龙格-库塔方法和$h=0.1$进行计算。这是一种生成标准正态分布的面积表的好方法。

8. 计算定积分$I(f)=\int_0^1 f(x)\,dx$可以看成求解常微分方程$\begin{cases}y'(x)=f(x)\\y(0)=0\end{cases}$的解函数在$x=1$处的值。试用几种常微分方程的算法建立数值积分公式，并求出它们的代数精度。

9. 考虑积分-常微分方程

$$y'=1.3y-0.25y^2-0.0001y\int_0^t y(\tau)\,d\tau$$

(1) 在区间[0,20]内，取$h=0.2$与$y(0)=250$，用欧拉方法和梯形公式求方程的近似解。提示：欧拉公式的一般步长表达式为

$$y_{i+1}=y_i+h\left(1.3y_i-0.25y_i^2-0.0001y_i\int_0^{t_i}y(\tau)\,d\tau\right)$$

若用梯形公式计算积分，则该表达式为

$$y_{i+1}=y_i+h(1.3y_i-0.25y_i^2-0.0001y_iT_i(h))$$

其中，$T_0(h)=0$，并且

$$T_i(h)=T_{i-1}(h)+\frac{h}{2}(y_{i-1}+y_i)\quad (i=1,2,\cdots)$$

(2) 在区间[0,20]内，取$h=0.2$与$y(0)=300$，用四阶龙格-库塔方法和梯形公式求方程的近似解。

10. 在化学反应中，一个A分子与一个B分子结合，形成一个C分子。发现t时刻C分子的浓度$y(t)$是一个初值问题

$$y'(t)=k(a-y)(b-y),\quad y(0)=0$$

其中，k为一正的常数，a和b分别是A分子和B分子的初始浓度。设$k=0.01$，$a=70$ mmol/L，$b=50$ mmol/L，取步长$h=0.5$，用四阶龙格-库塔方法计算区间[0,20]内的解。(精确解是$y(t)=350(1-\exp(-0.2t))/(7-5\exp(-0.2t))$)

11. 实验表明，放射性物质的衰变速率与未衰变放射性物质的量成正比，即

$$\frac{dy}{dt}=-ky$$

$y(t)$是t时刻未衰变的放射性物质总量，k是衰变常数。放射性物质的半衰期是初始物质衰减一半所需要的时间。已知^{14}C的半衰期是5 730年，用数值方法求解微分方程，并绘制^{14}C衰变率与时间的关系曲线。(微分方程的精确解是$y(t)=y(0)\exp(-kt)$)

12. 在打开降落伞之前，跳伞运动员受到的空气阻力与其下落速度的3/2次方成正比。设在时间区间[0,6]s内，跳伞运动员的下降速度满足微分方程

$$v'=32-0.032v^{3/2}$$

并且$v(0)=0$，试计算跳伞运动员在$t=1$s, 2s, 3s, 4s, 5s, 6s时的下落速度。

13. 设 $x(t)$ 和 $y(t)$ 分别表示 t 时刻兔子和狐狸的数量，捕食者-被捕食者模型表明，$x(t)$ 和 $y(t)$ 满足方程

$$\begin{cases} x'(t) = Ax(t) - Bx(t)y(t) \\ y'(t) = Cx(t)y(t) - Dy(t) \end{cases}$$

选取参数 $A=2, B=0.02, C=0.0002, D=0.8$，在区间[0,5]内，用步长 $h=0.2$，对初始状态为 $x(0)=3\,000$ 只兔子和 $y(0)=120$ 只狐狸的兔子与狐狸种群数量的变化进行计算。

14. 共振弹簧系统的动力学方程是

$$x''(t) + 25x(t) = 8\sin(5t), \quad x(0)=0, \ x'(0)=0$$

在区间[0,2]s 内，取步长 $h=0.05$s，对函数 $x(t)$ 进行数值求解。

15. 某个 RLC 电路的数学模型为

$$Q''(t) + 20Q'(t) + 125Q(t) = 9\sin(5t), \quad Q(0)=0, \quad Q'(0)=0$$

在区间[0,2]s 内，用步长 $h=0.05$s，对函数 $Q(t)$ 进行数值求解。（$I(t)=Q'(t)$ 是电流强度）

16. 已知受迫振动的方程是

$$x''(t) + 2\beta x'(t) + \omega_0^2 x(t) = \sin(\omega t) \quad (\beta < \omega_0)$$

取 $\omega_0=1, x(0)=1, x'(0)=0, \beta=0.1$ 或 0.4，步长 $\Delta t=0.5$，试用四阶龙格-库塔公式进行数值计算，并分别画出 $\omega=0.5,1,2$ 时的 $x(t)-t$ 及 $x'(t)-t$ 曲线。

17. 设单摆运动的角位移是 $x(t)$，则单摆运动的方程是

$$mlx''(t) = -mg\sin(x(t))$$

其中 m 为质量，l 为悬线长度，g 为重力加速度。取 $l=1.5$m，$g=10\text{m/s}^2$，步长 $h=0.05$s，$x(0)=0.3, x'(0)=0$，在区间[0,5]s 内，对函数 $x(t)$ 进行数值求解，并绘制 $x(t)-t$ 曲线。

第 8 章　解二阶偏微分方程的差分法

许多科学问题和工程问题的数学描述是偏微分方程(如哈密顿正则方程、麦克斯韦电磁场方程组、薛定谔方程、爱因斯坦引力场方程、扩散方程、波动方程、亥姆霍斯方程等),如同代数方程及常微分方程一样,寻求它们的解析解也是一件极其困难甚至不可能的事,因此用数值方法计算其近似解十分必要。本章主要介绍求解三大类(抛物型、双曲型和椭圆型)二阶偏微分方程的差分方法,通过计算实例研究了扩散现象和平行板电容器内的电势。

8.1　二阶偏微分方程的分类和解的特性

8.1.1　二阶偏微分方程的分类

考察二阶偏微分方程

$$a\frac{\partial^2 u}{\partial x^2}+b\frac{\partial^2 u}{\partial x\partial y}+c\frac{\partial^2 u}{\partial y^2}+d\frac{\partial u}{\partial x}+e\frac{\partial u}{\partial y}+fu=g(x,y) \tag{8.1}$$

若系数 a,b,c,d,e 及 f 是 x 和 y 的函数,则方程式(8.1)是线性偏微分方程;若系数 a,b 及 c 是 $x,y,u,\frac{\partial u}{\partial x}$ 及$\frac{\partial u}{\partial y}$ 的函数,则方程式(8.1)是拟线性偏微分方程。根据 a,b 及c 的大小关系,可以把方程式(8.1)分为以下三大类:

(1) 如果在点(x_0,y_0)处$\Delta(x_0,y_0)=b^2-4ac=0$,那么方程式(8.1)在点$(x_0,y_0)$处是抛物型的。二阶抛物型方程的标准型是

$$\frac{\partial^2 u}{\partial \xi^2}=h\left(\frac{\partial u}{\partial \xi},\frac{\partial u}{\partial \eta},\xi,\eta\right) \tag{8.2}$$

(2) 如果在点(x_0,y_0)处$\Delta(x_0,y_0)=b^2-4ac>0$,那么方程式(8.1)在点$(x_0,y_0)$处是双曲型的。二阶双曲型方程的标准型是

$$\frac{\partial^2 u}{\partial \xi^2}-\frac{\partial^2 u}{\partial \eta^2}=h\left(\frac{\partial u}{\partial \xi},\frac{\partial u}{\partial \eta},\xi,\eta\right) \tag{8.3}$$

也能表示成特征坐标形式

$$\frac{\partial^2 u}{\partial \xi\partial \eta}=h\left(\frac{\partial u}{\partial \xi},\frac{\partial u}{\partial \eta},\xi,\eta\right) \tag{8.4}$$

(3) 如果在点(x_0,y_0)处$\Delta(x_0,y_0)=b^2-4ac<0$时,那么方程式(8.1)在点$(x_0,y_0)$处是椭圆型的。二阶椭圆型方程的标准型是

$$\frac{\partial^2 u}{\partial \xi^2}+\frac{\partial^2 u}{\partial \eta^2}=h\left(\frac{\partial u}{\partial \xi},\frac{\partial u}{\partial \eta},\xi,\eta\right) \tag{8.5}$$

微分方程的类型在非奇异坐标变换下保持不变。双曲型方程和抛物型方程都属于时间发展型方程,它们通常用于描述随时间变化的非定常物理现象(如波动过程和热传导过程),而椭

圆型方程则用于描述不随时间变化的稳态物理现象(如静电场的电势)。时间发展型方程的适定性问题通常有两种,即初值问题和初边值问题,而椭圆型方程的适定性问题只能是边值问题。

8.1.2　二阶偏微分方程解的特性

这里通过几个典型例子介绍二阶偏微分方程解的基本特性。

1. 抛物型偏微分方程

一维热传导方程的初值问题是

$$\left.\begin{aligned} &\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2} \qquad (x\in\mathbf{R},\ t>0)\\ &u(x,0)=\varphi(x) \end{aligned}\right\} \tag{8.6}$$

如果函数 $\varphi(x)$ 有界并连续,则初值问题式(8.6) 的唯一解是

$$u(x,t)=\frac{1}{2\sqrt{\pi t}}\int_{-\infty}^{+\infty}\varphi(s)\,\mathrm{e}^{-(x-s)^2/(4t)}\,\mathrm{d}s \tag{8.7}$$

上式表明,解 $u(x,t)$ 在点 (x,t) 处的值依赖于初始状态函数 $\varphi(x)$ 在所有点的值,依赖程度随距离 $|x-s|$ 按二次负指数规律减小。这说明抛物型方程的信号传递速度是无限大,一点的扰动会很快传播到整个区域内,但在传播过程中信号的强度不断衰减。另外,解式(8.7) 满足极值原理:如果初始函数 $\varphi(x)$ 有界,那么方程的解 $u(x,t)$ 具有同样的上下界。

2. 双曲型偏微分方程

一维波动方程的初值问题是

$$\left.\begin{aligned} &\frac{\partial^2 u}{\partial t^2}-a^2\frac{\partial^2 u}{\partial x^2}=0 \qquad (a>0,x\in\mathbf{R},0<t<T)\\ &u(x,0)=\varphi_1(x) \qquad (\varphi_1\in C^2(\mathbf{R}))\\ &\frac{\partial u}{\partial t}(x,0)=\varphi_2(x) \qquad (\varphi_2\in C^1(\mathbf{R})) \end{aligned}\right\} \tag{8.8}$$

式中,$\mathbf{R}$ 为实数集合。上述初值问题的解由达朗贝尔(d'Alembert) 公式给出,即

$$u(x,t)=\frac{1}{2}(\varphi_1(x+at)+\varphi_1(x-at))+\frac{1}{2a}\int_{x-at}^{x+at}\varphi_2(s)\,\mathrm{d}s \tag{8.9}$$

可以看出,解 $u(x,t)$ 在点 (x,t) 处的值取决于初始函数 $\varphi_1(x)$ 在两点 $x\pm at$ 处的值和初始函数 $\varphi_2(x)$ 在区间 $[x-at,x+at]$ 内各点处的值。解式(8.9) 还能表示为

$$u(x,t)=F^+(x-at)+F^-(x+at) \tag{8.10}$$

其中,函数 $F^\pm(x\mp at)$ 是初值问题

$$\left.\begin{aligned} &\frac{\partial v}{\partial t}\pm a\frac{\partial v}{\partial x}=0\\ &v(x,0)=F^\pm(x) \end{aligned}\right\} \tag{8.11}$$

的解。式(8.10) 说明,经过时间 t 后,初始扰动 $F^\pm(x)$ 在空间的分布不变,只是整体沿 x 正方向或负方向移动了距离 at,而 a 就是初始扰动的传递速度,这就是波动的传播过程。

3. 椭圆型偏微分方程

拉普拉斯(Laplace) 方程(亦称调和方程) 为

$$\Delta u=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}=0 \qquad ((x,y,z)\in\Omega) \tag{8.12}$$

其中，Ω是单连通域。调和方程的解称为调和函数。满足方程式(8.12)的函数$u(x,y,z)$有下面的基本积分公式：

$$u(x,y,z)=-\frac{1}{4\pi}\int_{\partial\Omega}\left[u(x',y',z')\frac{\partial}{\partial n}\left(\frac{1}{r}\right)+\frac{1}{r}\frac{\partial u(x',y',z')}{\partial n}\right]\mathrm{d}s \tag{8.13}$$

式中，$\partial/\partial n$ 和 $\mathrm{d}s$ 分别表示区域Ω表面$\partial\Omega$的外法向梯度和面元，它们与空间点(x',y',z')有关，$r=\sqrt{(x-x')^2+(y-y')^2+(z-z')^2}$。式(8.13)表明，调和函数$u(x,y,z)$完全由区域边界上的值决定，这也说明椭圆型方程的定解问题只能是边值问题。调和函数的一个重要性质是极值原理：不恒等于常数的调和函数只能在区域的边界上取得最大值和最小值。

8.2 解偏微分方程的差分法

一般来说，有限差分法求解偏微分方程定解问题的过程主要包括三步：

(1) 将求解区域离散化，即对求解区域进行网格剖分。

用于空间一维初边值问题的网格由直线组

$$\left.\begin{aligned}&t=t_k=k\tau \quad (k=0,1,2,\cdots,K;\ K=[T/\tau])\\&x=x_i=ih \quad (i=0,1,2,\cdots,I;Ih=x_r-x_l)\end{aligned}\right\} \tag{8.14}$$

形成，所覆盖的区域是$\Omega=\{(x,t)\mid x_l\leqslant x\leqslant x_r,0\leqslant t\leqslant T\}$。式中$[\cdot]$表示取整，$h$和$\tau$分别是$x$方向和$t$方向的网格步长。

两组直线的交点称为网格节点，区域Ω边界上的网格节点称为边界节点，区域Ω内部的网格节点称为内部网格节点。网格剖分的过程又称为数值网格生成。它在微分方程的数值求解中非常重要，网格剖分质量直接影响数值求解的运算效率和计算精度。有限差分法是在网格节点上计算微分方程近似解的方法，又称为网格法。

(2) 将偏微分方程和定解条件离散化，建立代数方程组。

这一步就是用网格节点上的函数值近似表示微分方程中的偏导数及定解条件。如果h和τ足够小，那么函数$u(x,t)$在网格节点(x_i,t_k)处的各阶偏导数可以近似为

对x的一阶向前差商

$$\left.\frac{\partial u}{\partial x}\right|_{i,k}^{+}=\frac{u_{i+1,k}-u_{i,k}}{h} \tag{8.15}$$

对x的一阶向后差商

$$\left.\frac{\partial u}{\partial x}\right|_{i,k}^{-}=\frac{u_{i,k}-u_{i-1,k}}{h} \tag{8.16}$$

对x的二阶中心差商

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,k}=\frac{\left.\frac{\partial u}{\partial x}\right|_{i,k}^{+}-\left.\frac{\partial u}{\partial x}\right|_{i,k}^{-}}{h}=\frac{u_{i+1,k}-2u_{i,k}+u_{i-1,k}}{h^2} \tag{8.17}$$

对t的一阶向前差商

$$\left.\frac{\partial u}{\partial t}\right|_{i,k}=\frac{u_{i,k+1}-u_{i,k}}{\tau} \tag{8.18}$$

对t的二阶中心差商

$$\left.\frac{\partial^2 u}{\partial t^2}\right|_{i,k}=\frac{u_{i,k+1}-2u_{i,k}+u_{i,k-1}}{\tau^2} \tag{8.19}$$

其中,函数下标“i,k”表示函数在 $x=ih$,$t=k\tau$ 处取值,即 $u_{i,k}=u(ih,k\tau)$。

定解条件也可以进行相应的离散,如定解问题式(8.8)中的初始条件 $u(x,0)=\varphi_1(x)$ 能够离散为 $u_{i,0}=\varphi_1(ih)$。微分方程和定解条件离散后,就构成了关于 $u_{i,k}$ 的线性代数方程组。

(3) 解线性代数方程组,得到定解问题的数值解。

8.3　解抛物型方程的差分法

以一维热传导方程的初边值问题为例,介绍解抛物型方程的差分法。

设热传导方程的初边值问题是

$$\left.\begin{aligned}&\frac{\partial u}{\partial t}=a^2\frac{\partial^2 u}{\partial x^2} && (0<x<l,t>0)\\&u(0,t)=c_l,u(l,t)=c_r && (t>0)\\&u(x,0)=f(x) && (0<x<l)\end{aligned}\right\} \tag{8.20}$$

其中 a,l 是常数,定解区域 $\Omega=\{(x,t)\mid 0<x<l,t>0\}$,在 x-t 平面上是一个带状区域。

首先,对定解区域进行网格剖分。取正整数 I,定义空间步长 $h=l/I$,取时间步长为 τ。用两组直线

$$x=x_i=ih\quad(i=0,1,2,\cdots,I);\quad t=t_k=k\tau\quad(k=0,1,2,\cdots) \tag{8.21}$$

将区域 Ω 划分为矩形网格,网格节点坐标为 (x_i,t_k)。

然后,对热传导方程离散化。微分方程对区域内的每一点都成立,在网格节点 (x_i,t_k) 处,微分方程就是

$$\left.\frac{\partial u}{\partial t}\right|_{i,k}=a^2\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,k} \tag{8.22}$$

把差商公式(8.17)和公式(8.18)代入上式,得到差分公式(亦称差分格式)

$$u_{i,k+1}=(1-2\alpha)u_{i,k}+\alpha(u_{i-1,k}+u_{i+1,k})\quad(i=1,2,\cdots,I-1;k\geqslant 1) \tag{8.23}$$

其中

$$\alpha=\frac{a^2\tau}{h^2} \tag{8.24}$$

称为网格比。同时边值条件 $u(0,t)=c_l$ 与 $u(l,t)=c_r$ 和初值条件 $u(x,0)=f(x)$ 也离散化为

$$\left.\begin{aligned}&u_{0,k}=c_l,\ u_{I,k}=c_r\quad(k\geqslant 1)\\&u_{i,0}=f_i=f(ih)\quad(i=1,2,\cdots,I-1)\end{aligned}\right\} \tag{8.25}$$

联立差分方程式(8.23)和定解条件式(8.25),构成关于内部网格节点处函数值 $u_{i,k}(i=1,2,\cdots,I-1;k\geqslant 1)$ 的线性代数方程组,从中能够求出偏微分方程精确解 $u(x_i,t_k)$ 的近似值 $u_{i,k}(i=1,2,\cdots,I-1;k\geqslant 1)$,称为数值解。

根据差分方程式(8.23),可以由初始条件和边界条件逐次求出同一时刻各网格节点的 u 值。习惯上把同一时刻网格节点的 u 的全体称为一层,而计算则是逐层推进的。计算过程中层间网格节点的关系如图 8.1 所示,$k+1$ 层第 i 个节点处的 u 值由 k 层第 $i-1,i,i+1$ 三个节

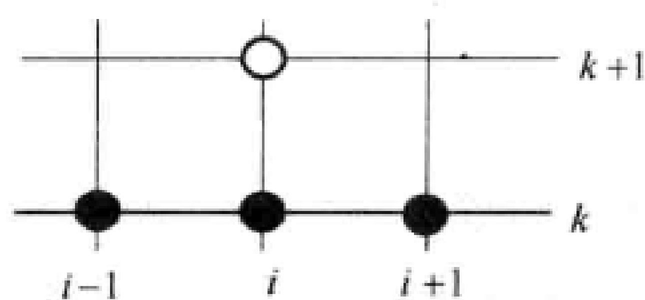


图 8.1

点处的 u 值推算出来。由于只涉及四个网格节点，并且当第 k 层上所有网格节点处的 $u_{i,k}(i=0,1,2,\cdots,I)$ 已知时，不需要求解线性方程组，便可直接从差分方程式(8.23) 计算出 $k+1$ 层上所有内部网格节点处的 $u_{i,k+1}(i=1,2,\cdots,I-1)$ 值，所以称方程式(8.23) 为四点显式差分格式。

如果定义四点显式差分格式的解向量 $\boldsymbol{u}_k$ 为

$$\boldsymbol{u}_k=[u_{1,k}\quad u_{2,k}\quad\cdots\quad u_{I-1,k}]^{\mathrm{T}}\quad(k=1,2,3,\cdots)\tag{8.26}$$

并定义 $I-1$ 阶三对角矩阵 $\mathbf{A}$ 为

$$\mathbf{A}=\begin{bmatrix}1-2\alpha & \alpha & & \\ \alpha & \ddots & \ddots & \\ & \ddots & \ddots & \alpha \\ & & \alpha & 1-2\alpha\end{bmatrix}\tag{8.27}$$

则四点显式差分格式又能表示为向量乘积的形式，即

$$\boldsymbol{u}_{k+1}=\boldsymbol{A}\boldsymbol{u}_k\quad(k=0,1,2,\cdots)\tag{8.28}$$

并且

$$\boldsymbol{u}_0=[f(h)\quad f(2h)\quad\cdots\quad f((I-1)h)]^{\mathrm{T}}\tag{8.29}$$

这就是说，对于四点显式差分格式，从第 k 层解向量推算第 $k+1$ 层解向量只需做一次简单的矩阵乘法。

在网格节点 (x_i,t_k) 处，用差分公式(8.23) 近似偏微分方程式(8.22)，产生的截断误差是

$$R_{i,k}=\frac{\tau}{2}\frac{\partial^2 u}{\partial t^2}(x_i,\eta_k)-\frac{a^2h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i,t_k)\tag{8.30}$$

其中，$\xi_i\in(x_{i-1},x_{i+1})$，$\eta_k\in(t_k,t_{k+1})$。这个误差称为差分方程式(8.23) 的局部截断误差，它定量地描述了差分方程近似偏微分方程的程度。要使误差表达式(8.30) 成立，偏微分方程的解 u 必须关于 x 有四阶连续偏导数，关于 t 有二阶连续偏导数。此时，差分方程的精度关于 x 为二阶，关于 t 为一阶。

如果在逐层向上计算时，初始误差不被扩大，则称差分格式是数值稳定的。可以证明，差分公式(8.23) 数值稳定的充分条件是

$$\alpha=\frac{a^2\tau}{h^2}\leqslant\frac{1}{2}\tag{8.31}$$

例 8.1 用四点显式差分格式求解热传导方程的定解问题

$$\begin{cases}\dfrac{\partial u}{\partial t}=\dfrac{\partial^2 u}{\partial x^2} & (0<x<1,t>0)\\ u(0,t)=u(1,t)=0 & (t>0)\\ u(x,0)=\sin(\pi x) & (0<x<1)\end{cases}$$

解 该定解问题的解析解为 $u(x,t)=\mathrm{e}^{-\pi^2 t}\sin(\pi x)$。并且 $a=1,l=1$，取 $h=0.1,\tau=0.000\,5$。则网格比

$$\alpha=\frac{a^2\tau}{h^2}=0.05$$

$$\text{三对角矩阵 } \boldsymbol{A}=\begin{bmatrix} 0.9 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.05 & 0.9 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.05 & 0.9 & 0.05 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0.9 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0.9 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.05 & 0.9 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.05 & 0.9 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0.9 & 0.05 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0.9 \end{bmatrix}$$

初始向量是

$$\boldsymbol{u}_0=[\sin(0.1\pi)\quad \sin(0.2\pi)\quad \sin(0.3\pi)\quad \sin(0.4\pi)\quad \sin(0.5\pi)\quad \sin(0.6\pi)\quad \sin(0.7\pi)\quad \sin(0.8\pi)\quad \sin(0.9\pi)]^{\mathrm{T}}$$

根据公式(8.28)，各层解向量是

$$\boldsymbol{u}_1=\boldsymbol{A}\boldsymbol{u}_0,\quad \boldsymbol{u}_2=\boldsymbol{A}\boldsymbol{u}_1=\boldsymbol{A}^2\boldsymbol{u}_0,\quad \boldsymbol{u}_3=\boldsymbol{A}\boldsymbol{u}_2=\boldsymbol{A}^3\boldsymbol{u}_0,\quad \cdots\cdots$$

即

$$\boldsymbol{u}_k=\boldsymbol{A}^k\boldsymbol{u}_0\quad (k=1,2,3,\cdots)$$

求出的几组解向量见表 8.1。

表　8.1

	$x_1=0.1$	$x_2=0.2$	$x_3=0.3$	$x_4=0.4$	$x_5=0.5$
$t_0=0.000$	0.309 0	0.587 8	0.809 0	0.951 1	1.000 0
$t_{100}=0.05$	0.189 2	0.359 9	0.495 3	0.582 3	0.612 2
$t_{200}=0.10$	0.115 8	0.220 3	0.303 2	0.356 5	0.374 8
$t_{400}=0.20$	0.043 4	0.082 6	0.113 7	0.133 6	0.140 5
$t_{800}=0.40$	0.006 1	0.011 6	0.016 0	0.018 8	0.019 7
$t_{1\,000}=0.50$	0.002 3	0.004 3	0.006 0	0.007 0	0.007 4

程序(8.1)　用差分法解抛物型方程的 MATLAB 程序。

程序任务：用差分法解一维抛物型方程的初边值问题(8.20)。

```
function U = DifParab(fun, cl, cr,a, L, T, I, K)
% 输入:fun—— 初始条件函数 u(x,0)
%       cl,cr—— 边界条件 u(0,t) = cl 及 u(L,t) = cr
%       a—— 抛物型方程 u_t = a^2 u_xx 中的常数 a
%       L,T—— 空间范围[0, L] 和时间范围[0, T] 的上边界
%       I,K—— 区间[0,L] 的划分数为 I,区间[0, T] 的划分数为 K
% 输出:U—— 方程的解矩阵(K + 1) × (I + 1)
h = L/I; t = T/K;              % 计算空间步长和时间步长
r = a^2 * t/h^2;               % 计算网格比,公式(8.24)
U = zeros(I + 1, K + 1);       % 初始化解矩阵
```

```
U(1, 1:(K+1)) = cl; U((I+1), 1:(K+1)) = cr;          % 赋予边界条件
U(2:I, 1) = feval(fun, h:h:(I-1)*h)';                 % 赋予初始条件
for k = 2:K
  for i = 2:I
      U(i,k) = (1-2*r)*U(i,k-1)+r*(U(i-1,k-1)+U(i+1,k-1));% 公式(8.23)
  end
end
U = U';
```

例 8.2 用程序(8.1)解下列初边值问题：

$$
\begin{cases}
\dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2} & (0 < x < 1, 0 < t < 0.2) \\
u(0,t) = u(1,t) = 0 & (0 < t < 0.2) \\
u(x,0) = 4\exp\left[-\dfrac{(x-0.5)^2}{0.2^2}\right] & (0 < x < 1)
\end{cases}
$$

解 在MATLAB命令窗口输入：

```
>> fun=inline('4*exp(-(x-0.5).^2/0.2^2)','x');        % 建立初始条件函数
>> U=DifParab(fun, 0, 0, 1, 1, 0.20, 8, 30);          % 调用程序(8.1)计算
>> U                                                    % 输出计算结果
```

输出结果(部分)见表8.2。

表 8.2

	$x_0=0.000$	$x_1=0.125$	$x_2=0.250$	$x_3=0.375$	$x_4=0.500$
$t_0=0.000$	0.000 0	0.118 9	0.838 4	2.706 5	4.000 0
$t_3=0.020$	0.000 0	0.690 7	1.394 5	1.994 2	2.216 6
$t_6=0.040$	0.000 0	0.644 4	1.206 2	1.590 0	1.732 0
$t_9=0.060$	0.000 0	0.536 1	0.991 4	1.297 9	1.404 8
$t_{12}=0.080$	0.000 0	0.439 0	0.811 5	1.060 3	1.147 9
$t_{15}=0.100$	0.000 0	0.359 1	0.663 4	0.866 9	0.938 2

```
>> meshz(U)      % 绘制三维示意图
```

输出结果如图8.2所示。

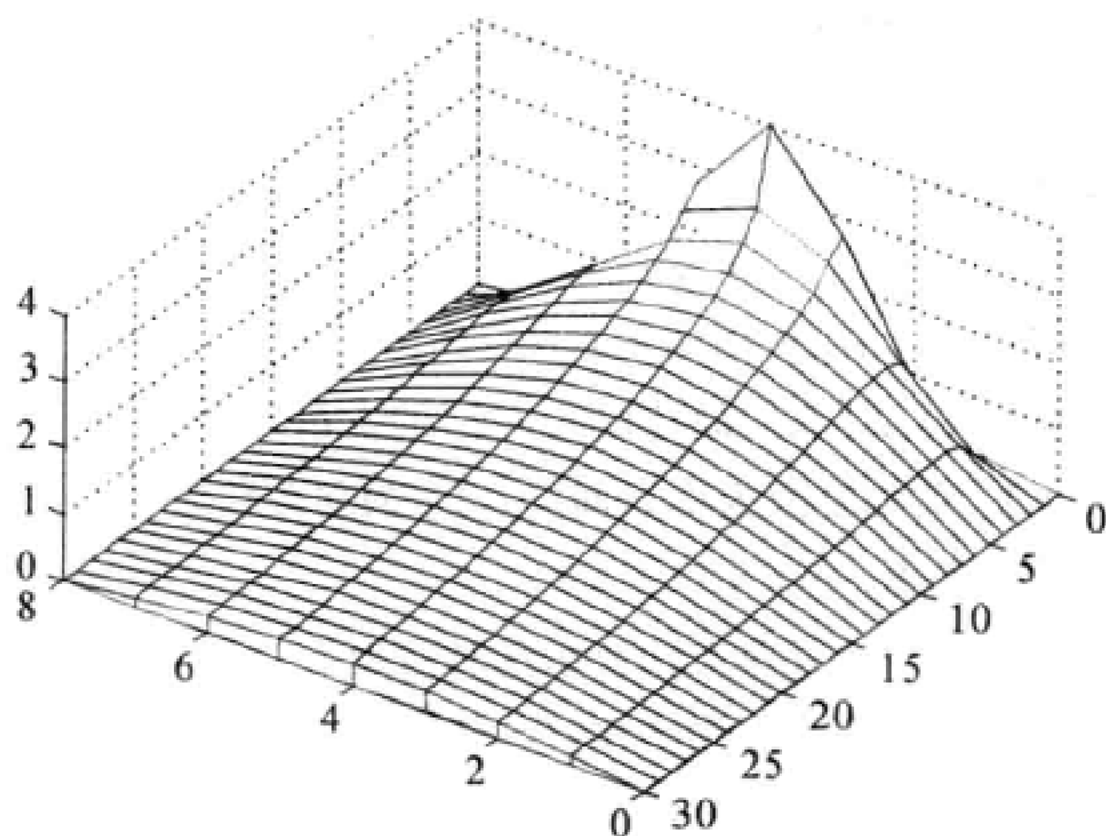


图 8.2

8.4　解双曲型方程的差分法

以一维波动方程的初边值问题为例，介绍解双曲型方程的差分法。

均匀弦线自由振动的方程为

$$\frac{\partial^2 u}{\partial x^2}-\frac{1}{v^2}\frac{\partial^2 u}{\partial t^2}=0 \quad (0<x<l, 0<t<T) \tag{8.32}$$

其中 v 是波速，l 是弦线长度，T 表示一段时间。定解区域 $\Omega=\{(x,t)\mid 0<x<l, 0<t<T\}$，在 $x-t$ 平面上是一个矩形区域。

对定解区域进行网格剖分。取正整数 I 和 K，空间步长 $h=l/I$，时间步长 $\tau=T/K$。用两组直线

$$x=x_i=ih \quad (i=0,1,2,\cdots,I);\quad t=t_k=k\tau \quad (k=0,1,2,\cdots,K) \tag{8.33}$$

将区域 Ω 划分为矩形网格，网格节点坐标为 (x_i,t_k)。

对偏微分方程式(8.32)离散化。微分方程对区域内的每一点都成立，在网格节点 (x_i,t_k) 处，微分方程就是

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,k}-\frac{1}{v^2}\left.\frac{\partial^2 u}{\partial t^2}\right|_{i,k}=0 \tag{8.34}$$

把二阶中心差商公式(8.17)和公式(8.19)代入式(8.34)，得到差分公式

$$u_{i,k+1}=2(1-\lambda^2)u_{i,k}+\lambda^2(u_{i+1,k}+u_{i-1,k})-u_{i,k-1} \tag{8.35}$$

其中

$$\lambda=\frac{v\tau}{h} \tag{8.36}$$

利用差分公式(8.35)，可以由第 i 个节点处 $k-1$ 时刻及第 $i-1,i,i+1$ 三个节点处 k 时刻的 u 值计算出第 i 个节点处 $k+1$ 时刻的 u 值，相关节点的关系如图 8.3 所示。

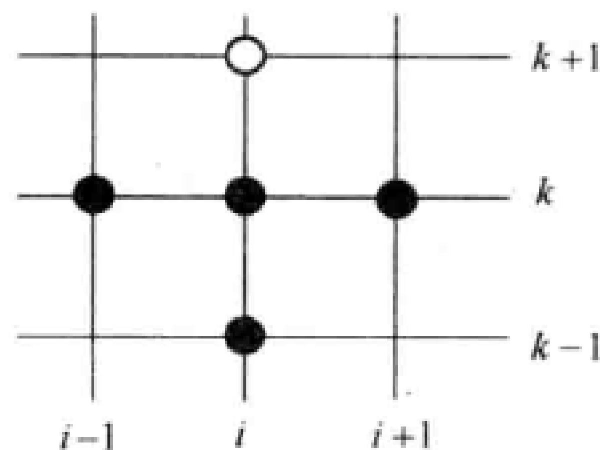


图　8.3

该类问题的初始条件一般为

$$\left.\begin{aligned} &u(x,0)=\varphi(x)\\ &\frac{\partial u}{\partial t}(x,0)=\psi(x)\end{aligned}\quad (0\leqslant x\leqslant l)\right\} \tag{8.37}$$

边界条件一般为

$$\left.\begin{aligned} &u(0,t)=g(t)\\ &u(l,t)=f(t)\end{aligned}\quad (0\leqslant t\leqslant T)\right\} \tag{8.38}$$

初始条件给出了初始时刻弦线上各点的位移和速度，边界条件给出了弦线两端位移随时间的变化。式(8.37)中速度初始条件的差分公式一般有以下两种：

(1) 使用向前的一阶差商式(8.18)，即 $(\partial u/\partial t)_{i,0}=(u_{i,1}-u_{i,0})/\tau$，得到初始条件的第一种差分公式

$$u_{i,1}=u_{i,0}+\tau\psi(ih)=\varphi(ih)+\tau\psi(ih) \quad (i=0,1,2,\cdots,I) \tag{8.39}$$

(2) 使用中心一阶差商公式

$$\left.\frac{\partial u}{\partial t}\right|_{i,0}=\frac{u_{i,1}-u_{i,-1}}{2\tau} \quad (i=0,1,2,\cdots,I) \tag{8.40}$$

得到初始条件的第二种差分公式

$$u_{i,1} - u_{i,-1} = 2\tau\psi(ih) \tag{8.41}$$

在差分方程式(8.35)中令 $k=0$,有

$$u_{i,1} = 2(1-\lambda^2)u_{i,0} + \lambda^2(u_{i+1,0} + u_{i-1,0}) - u_{i,-1} \tag{8.42}$$

以上两式联立,求出

$$u_{i,1} = (1-\lambda^2)u_{i,0} + \frac{\lambda^2}{2}(u_{i+1,0} + u_{i-1,0}) + \tau\psi(ih) =$$

$$(1-\lambda^2)\varphi(ih) + \frac{\lambda^2}{2}(\varphi((i+1)h) + \varphi((i-1)h)) + \tau\psi(ih) \tag{8.43}$$

这样,一维波动方程初边值问题的差分公式可以有以下两种:

(1) 第一种差分公式

$$\left.\begin{array}{ll} u_{i,k+1} = 2(1-\lambda^2)u_{i,k} + \lambda^2(u_{i+1,k} + u_{i-1,k}) - u_{i,k-1} & (i=1,2,\cdots,I-1;k=1,2,\cdots,K-1) \\ u_{i,0} = \varphi(ih) & (i=0,1,2,\cdots,I) \\ u_{i,1} = \varphi(ih) + \tau\psi(ih) & (i=0,1,2,\cdots,I) \\ u_{0,k} = g(k\tau) & (k=0,1,2,\cdots,K) \\ u_{I,k} = f(k\tau) & (k=0,1,2,\cdots,K) \end{array}\right\} \tag{8.44}$$

(2) 第二种差分公式

$$\left.\begin{array}{ll} u_{i,k+1} = 2(1-\lambda^2)u_{i,k} + \lambda^2(u_{i+1,k} + u_{i-1,k}) - u_{i,k-1} & (i=1,2,\cdots,I-1;k=1,2,\cdots,K-1) \\ u_{i,0} = \varphi(ih) & (i=0,1,2,\cdots,I) \\ u_{i,1} = (1-\lambda^2)\varphi(ih) + \dfrac{\lambda^2}{2}(\varphi((i+1)h) + \varphi((i-1)h)) + \tau\psi(ih) & (i=0,1,2,\cdots,I-1) \\ u_{0,k} = g(k\tau) & (k=0,1,2,\cdots,K) \\ u_{I,k} = f(k\tau) & (k=0,1,2,\cdots,K) \end{array}\right\} \tag{8.45}$$

第二种差分公式通常被采用,它比第一种差分公式有较高的精度。两种差分公式收敛且稳定的条件是

$$\lambda = \frac{v\tau}{h} \leqslant 1 \tag{8.46}$$

例 8.3 用第一种差分公式计算波动方程的混合问题

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} - \dfrac{\partial^2 u}{\partial t^2} = 0 & (0 < x < 1, t > 0) \\ u(x,0) = \sin(\pi x), \quad \dfrac{\partial u}{\partial t}(x,0) = x(1-x) & (0 \leqslant x \leqslant 1) \\ u(0,t) = u(1,t) = 0 & (t > 0) \end{cases}$$

取 $\lambda=1, h=0.2$,分别计算 $k=1,2,3,4$ 层的数值解。

解 根据 $\lambda=1, h=0.2, l=1$ 及 $v=1$,有

$$\tau = 0.2, \quad I = 5$$

差分公式(8.44)具体为

$$\begin{cases} u_{i,k+1} = u_{i+1,k} + u_{i-1,k} - u_{i,k-1} & (i=1,2,3,4;k=1,2,\cdots) \\ u_{i,0} = \sin(0.2i\pi) & (i=0,1,2,3,4,5) \\ u_{i,1} = \sin(0.2i\pi) + 0.04i(1-0.2i) & (i=0,1,2,3,4,5) \\ u_{0,k} = 0 & (k=0,1,2,\cdots) \\ u_{5,k} = 0 & (k=0,1,2,\cdots) \end{cases}$$

计算结果列入表 8.3 中。

表　8.3

	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
$k=0$	0	0.587 8	0.951 1	0.951 1	0.587 8	0
$k=1$	0	0.619 8	0.999 1	0.999 1	0.619 8	0
$k=2$	0	0.411 3	0.667 8	0.667 8	0.411 3	0
$k=3$	0	0.048 0	0.080 0	0.080 0	0.048 0	0
$k=4$	0	−0.331 3	−0.539 8	−0.539 8	−0.331 3	0

程序(8.2)　用差分法解双曲型方程的 MATLAB 程序。

程序任务：用差分法公式(8.45)解一维双曲型方程的初边值问题式(8.32)、式(8.37)及式(8.38)。

```
function U = DifHyper(dfun, vfun, lfun, rfun, v, L, T, I, K)
% 输入：dfun—— 初始条件位移函数 u(x,0)
%       vfun—— 初始条件速度函数 u_t(x,0)
%       lfun—— 左边界条件函数 u(0,t)
%       rfun—— 右边界条件函数 u(L,t)
%       v—— 双曲型方程 u_tt = v^2 u_xx 中的波速 v
%       L,T—— 空间范围[0, L]和时间范围[0, T]的上界
%       I,K——[0,L]的划分区间数为 I,[0, T]的划分区间数为 K
% 输出：U—— 方程的解矩阵(K+1)×(I+1)
h = L/I; t = T/K;            % 计算空间步长和时间步长
r = v * t/h;                 % 计算参数 λ,公式(8.46)
U = zeros(I+1, K+1);         % 初始化解矩阵
for i = 2:I
    U(i, 1) = feval(dfun, (i-1) * h);          % 计算 U 的第一列
    U(i, 2) = (1-r^2) * feval(dfun, (i-1) * h)+(r^2/2) * (feval(dfun, i * h)+feval(dfun, (i-2) * h))
            + t * feval(vfun, (i-1) * h);  % 计算 U 的第二列
end
for k = 1:(K+1)
    U(1,k) = feval(lfun, (k-1) * t);           % 计算 U 的第一行(左边界条件)
    U((I+1),k) = feval(rfun, (k-1) * t);  % 计算 U 的第(I+1)行(右边界条件)
end
for k = 3:K
    for i = 2:I
```

```
        U(i,k) = 2*(1-r2)*U(i,k-1)+r2*(U(i-1,k-1)+U(i+1,k-1))-U(i,k-2);
                                                    % 计算 U 的第 i 行,公式(8.35)
    end
end
U = U';
```

例 8.4 应用程序(8.2)求解下列初边值问题:

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} - \dfrac{1}{4}\dfrac{\partial^2 u}{\partial t^2} = 0 & (0 < x < 1, 0 < t < 0.5) \\ u(x,0) = \begin{cases} x \\ 1.5(1-x) \end{cases} & \begin{matrix} (0 \leqslant x \leqslant 3/5) \\ (3/5 \leqslant x \leqslant 1) \end{matrix} \\ u_t(x,0) = 0 & (0 \leqslant x \leqslant 1) \\ u(0,t) = u(1,t) = 0 & (0 < t < 0.5) \end{cases}$$

取 $h = 0.1, \tau = 0.05$。

解 编制并存储函数文件 dfun $= u(x,0)$,vfun $=0$,lfun $=0$,rfun $=0$ 后,在 MATLAB 命令窗口输入:

```
>> U = DifHyper('dfun', 'vfun', 'lfun', 'rfun', 2, 1, 0.5, 10, 10)
                                            % 调用程序(8.2) 计算
```

输出结果(部分) 见表 8.4。

表 8.4

	$x_1 = 0.1$	$x_3 = 0.3$	$x_5 = 0.5$	$x_7 = 0.7$	$x_9 = 0.9$
$t_2 = 0.1$	0.100 0	0.300 0	0.375 0	0.325 0	0.150 0
$t_4 = 0.2$	0.100 0	0.175 0	0.125 0	0.075 0	0.025 0
$t_6 = 0.3$	−0.025 0	−0.075 0	−0.125 0	−0.175 0	−0.100 0
$t_8 = 0.4$	−0.150 0	−0.325 0	−0.375 0	−0.300 0	−0.100 0
$t_{10} = 0.5$	−0.150 0	−0.450 0	−0.500 0	−0.300 0	−0.100 0

8.5 解椭圆型方程的差分法

常见的椭圆型方程包括拉普拉斯(Laplace) 方程、泊松(Possion) 方程和亥姆霍兹(Helmholtz) 方程。二维空间的拉普拉斯方程、泊松方程和亥姆霍兹方程分别是

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{8.47}$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = g(x,y) \tag{8.48}$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x,y)u = g(x,y) \tag{8.49}$$

通常情况下,函数 $u(x,y)$ 或其法向导数在矩形区域 $\Omega = \{(x,y) \mid 0 \leqslant x \leqslant a, 0 \leqslant y \leqslant b\}$ 边

界上的值是已知的，用差分方法就可以计算出方程的数值解。

先建立求解拉普拉斯方程式(8.47)的差分法。对定解区域进行网格剖分。取正整数 I 和 K，设 x 方向和 y 方向的步长均是 $h=a/I=b/K$。用两组直线

$$x=x_i=ih \quad (i=0,1,2,\cdots,I)\,; \quad y=y_k=kh \quad (k=0,1,2,\cdots,K) \tag{8.50}$$

将区域 Ω 划分为正方形网格，网格节点坐标为 (x_i,y_k)。

对偏微分方程式(8.47)离散化。微分方程对区域内的每一点都成立，在内部网格节点 (x_i,y_k) 处，微分方程就是

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,k}+\left.\frac{\partial^2 u}{\partial y^2}\right|_{i,k}=0 \tag{8.51}$$

用二阶中心差商代替二阶偏导数，得到差分公式

$$u_{i+1,k}+u_{i-1,k}+u_{i,k+1}+u_{i,k-1}-4u_{i,k}=0 \quad (i=1,2,\cdots,I-1;k=1,2,\cdots,K-1) \tag{8.52}$$

其中，$u_{i,k}=u(ih,kh)$。差分公式(8.52)的精度为 $o(h^2)$。利用差分公式(8.52)，可以由节点 $(i+1,k)$，$(i-1,k)$，$(i,k+1)$ 和 $(i,k-1)$ 处的函数值计算出节点 (i,k) 处的函数值，相关节点的关系如图 8.4 所示。

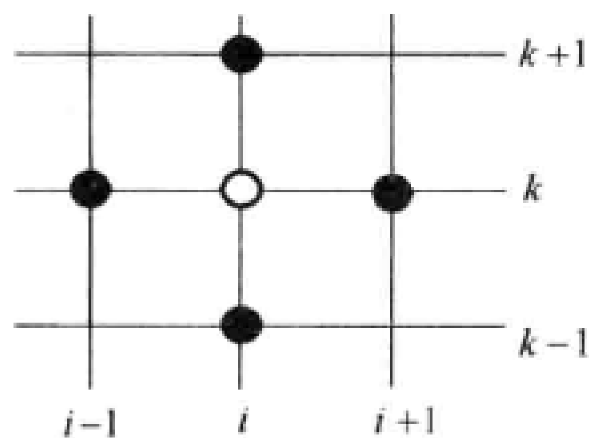


图　8.4

若边界条件是已知函数 $u(x,y)$ 在区域 Ω 边界上的值，即函数 $u(0,y)$ 与 $u(a,y)$，$u(x,0)$ 与 $u(x,b)$ 已知，则边界网格节点(共 $2(I+K)$ 个)处的函数值已知，即

$$u_{0,k}=u(0,kh) \quad (k=1,2,\cdots,K-1) \quad (\text{左边界}) \tag{8.53}$$

$$u_{I,k}=u(a,kh) \quad (k=1,2,\cdots,K-1) \quad (\text{右边界}) \tag{8.54}$$

$$u_{i,0}=u(ih,0) \quad (i=0,1,\cdots,I) \quad (\text{下边界}) \tag{8.55}$$

$$u_{i,K}=u(ih,b) \quad (i=0,1,\cdots,I) \quad (\text{上边界}) \tag{8.56}$$

若边界条件是已知函数 $u(x,y)$ 在区域 Ω 边界上的法向导数值，即诺埃曼(Neumann)边界条件。比如法向导数等于零，即

$$\left.\frac{\partial u}{\partial n}\right|_{\Omega\text{边界}}=0 \tag{8.57}$$

以区域的右边界 $(x=x_I=a)$ 为例进行分析。右边界条件是

$$\frac{\partial u}{\partial x}(x_I,y_k)=0 \quad (k=1,2,\cdots,K-1)$$

在边界节点 (x_I,y_k) 处应用差分方程式(8.52)和中心微分公式，有

$$u_{I+1,k}+u_{I-1,k}+u_{I,k+1}+u_{I,k-1}-4u_{I,k}=0$$

$$\frac{\partial u}{\partial x}(x_I,y_k)\approx\frac{u_{I+1,k}-u_{I-1,k}}{2h}=0$$

从以上两式可得

$$2u_{I-1,k}+u_{I,k+1}+u_{I,k-1}-4u_{I,k}=0 \quad (\text{右边界}) \tag{8.58}$$

它把 $u_{I,k}$ 与相邻节点的函数值 $u_{I-1,k}$，$u_{I,k+1}$ 和 $u_{I,k-1}$ 联系了起来。同样可以得出其他三段边界上的类似公式

$$2u_{1,k}+u_{0,k+1}+u_{0,k-1}-4u_{0,k}=0 \quad (\text{左边界}) \tag{8.59}$$

$$2u_{i,1}+u_{i+1,0}+u_{i-1,0}-4u_{i,0}=0 \quad (\text{下边界}) \tag{8.60}$$

$$2u_{i,K-1}+u_{i+1,K}+u_{i-1,K}-4u_{i,K}=0\ (\text{上边界}) \tag{8.61}$$

边界节点处函数值的联系如图 8.5 所示。

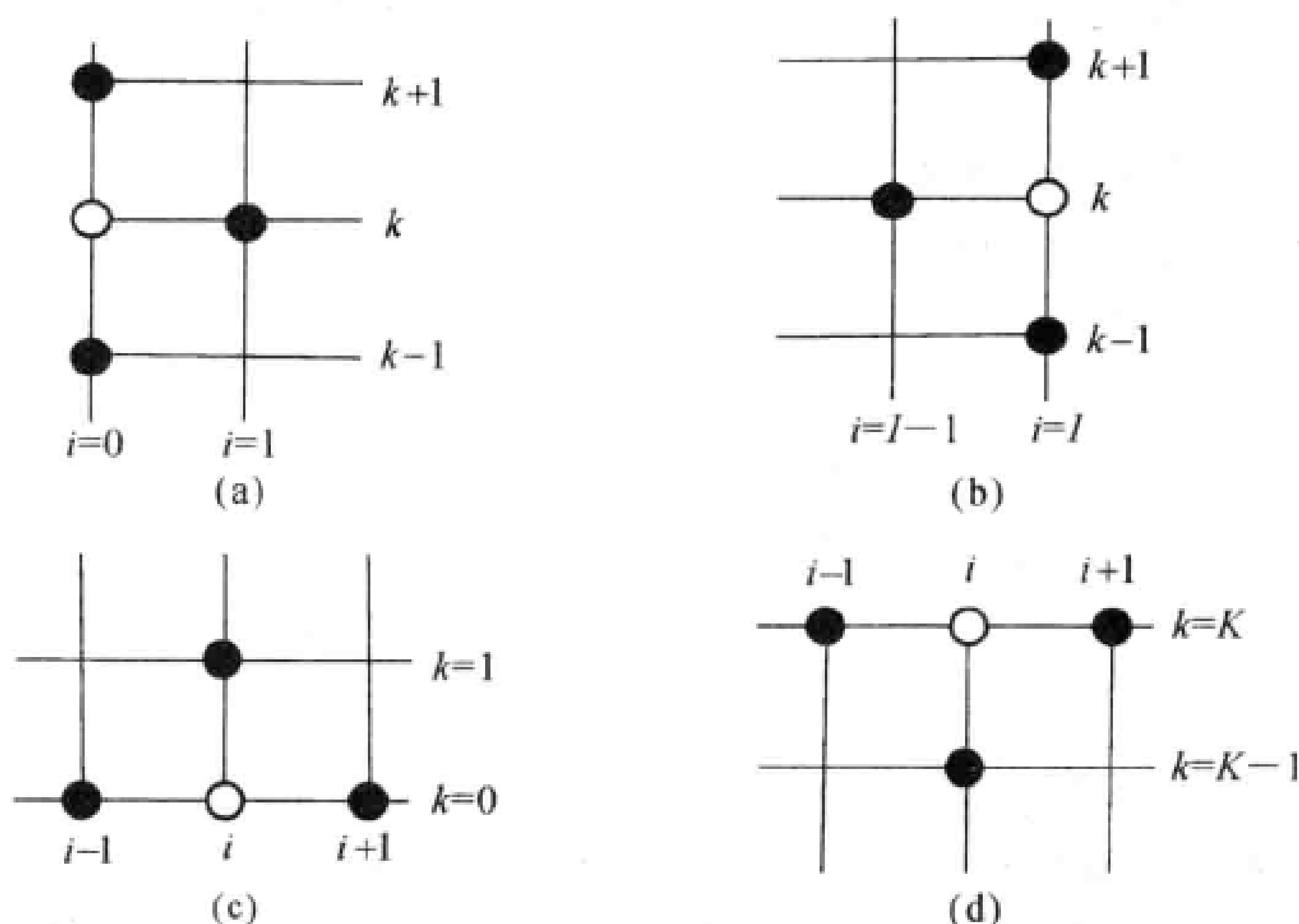


图 8.5

(a) 左边界； (b) 右边界； (c) 下边界； (d) 上边界

区域Ω的内部网格节点共$(I-1)(K-1)$个，边界网格节点共$2(I+K)$个。内部网格节点处的函数值满足差分公式(8.52)，边界网格结点处的函数值满足式(8.53)～式(8.56)或式(8.58)～式(8.61)。利用这些关系，可以建立以内部网络节点处的函数值为未知量的线性方程组，从中解出内部网络节点处的函数值，就得到了定解问题的数值解。

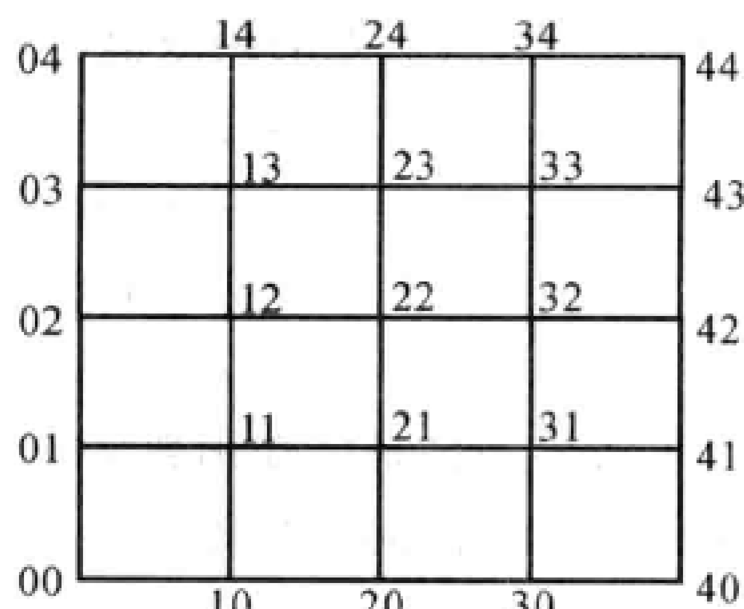


图 8.6

考察一个有 5×5 个网格节点的方形区域，如图 8.6 所示。网格节点的编号标在图中，设函数在边界网格节点处的值均为已知。内部节点共有 9 个，对这 9 个节点应用差分公式建立线性方程组。其中节点 22 不与边界节点相邻，此处的方程为齐次方程，其余 8 个节点都与边界节点相邻，关于它们的方程均为非齐次方程。所建立的线性方程组是

$$\begin{bmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{bmatrix} = \begin{bmatrix} -u_{1,0}-u_{0,1} \\ u_{2,0} \\ -u_{3,0}-u_{4,1} \\ -u_{0,2} \\ 0 \\ -u_{4,2} \\ -u_{0,3}-u_{1,4} \\ -u_{2,4} \\ -u_{3,4}-u_{4,3} \end{bmatrix} \tag{8.62}$$

从中能够解出内部网格节点处的函数值 $u_{i,k}(i,k=1,2,3)$。

用解线性方程组的方法求解拉普拉斯方程定解问题的缺点是占用存储空间大，计算时间长，因为每一个内部网格节点都会涉及一个方程。如果用迭代的方法，则这些缺点就可以克服。对于拉普拉斯方程及边界条件式(8.53) ~ 式(8.56) 构成的定解问题，可以从差分公式(8.52) 建立迭代公式

$$u_{i,k} = u_{i,k} + r_{i,k} \quad (i = 1,2,\cdots,I-1; k = 1,2,\cdots,K-1) \tag{8.63}$$

式中

$$r_{i,k} = \frac{u_{i+1,k} + u_{i-1,k} + u_{i,k+1} + u_{i,k-1} - 4u_{i,k}}{4} \tag{8.64}$$

内部网格节点处的迭代初值选为边界网格节点处函数值的平均值。这样的选定是科学的，也是可行的，因为数学上已经证明，函数在定解区域的最大值和最小值均在区域的边界上。用式(8.63) 对所有内部网格节点连续进行迭代计算，直到右边余项 $r_{i,k}$ 的绝对值满足精度要求，即 $|r_{i,k}| < \varepsilon$。

如果在迭代计算过程中，随时用上一步计算出来的新值替代旧值，甚至每次计算新值也替换成新值与旧值的"组合"，则可以得到松弛迭代公式

$$u_{i,k} = u_{i,k} + \omega r_{i,k} = (1-\omega)u_{i,k} + \frac{\omega}{4}(u_{i+1,k} + u_{i-1,k} + u_{i,k+1} + u_{i,k-1}) \tag{8.65}$$

其中，$0 < \omega < 2$。当 $\omega < 1$ 时称为"低松弛"，当 $\omega > 1$ 时称为"超松弛"。ω 的优化取值公式是

$$\omega = \frac{4}{2 + \sqrt{4 - (\cos(\pi/I) + \cos(\pi/K))^2}} \tag{8.66}$$

此时，诺埃曼边界条件的差分公式(8.58) ~ 公式(8.61) 分别改写为

$$u_{I,k} = u_{I,k} + \omega\left(\frac{2u_{I-1,k} + u_{I,k+1} + u_{I,k-1} - 4u_{I,k}}{4}\right) \quad (\text{右边界}) \tag{8.67}$$

$$u_{0,k} = u_{0,k} + \omega\left(\frac{2u_{1,k} + u_{0,k+1} + u_{0,k-1} - 4u_{0,k}}{4}\right) \quad (\text{左边界}) \tag{8.68}$$

$$u_{i,0} = u_{i,0} + \omega\left(\frac{2u_{i,1} + u_{i+1,0} + u_{i-1,0} - 4u_{i,0}}{4}\right) \quad (\text{下边界}) \tag{8.69}$$

$$u_{i,K} = u_{i,K} + \omega\left(\frac{2u_{i,K-1} + u_{i+1,K} + u_{i-1,K} - 4u_{i,K}}{4}\right) \quad (\text{上边界}) \tag{8.70}$$

采用相同的方法建立的计算泊松方程 $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = g(x,y)$ 和亥姆霍兹方程 $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 + f(x,y)u = g(x,y)$ 数值解的差分迭代公式分别是

$$u_{i,k} = (1-\omega)u_{i,k} + \frac{\omega}{4}(u_{i+1,k} + u_{i-1,k} + u_{i,k+1} + u_{i,k-1} - h^2 g_{i,k}) \tag{8.71}$$

$$u_{i,k} = (1-\omega)u_{i,k} + \omega\left(\frac{u_{i+1,k} + u_{i-1,k} + u_{i,k+1} + u_{i,k-1} - h^2 g_{i,k}}{4 - h^2 f_{i,k}}\right) \tag{8.72}$$

其中，$g_{i,k} = g(ih, kh)$，$f_{i,k} = f(ih, kh)$。

程序(8.3)　用松弛迭代法解二维拉普拉斯方程的 MATLAB 程序。

程序任务：用松弛迭代公式(8.65) 解二维拉普拉斯方程的定解问题。

```
function U = DifLapl(dfun, ufun, lfun, rfun, a, b, h, ep, Nm)
% 输入：dfun—— 下边界条件函数 u(x,0)
%       ufun—— 上边界条件函数 u(x,b)
%       lfun—— 左边界条件函数 u(0,y)
```

```
%         rfun—— 右边界条件函数 u(a,y)
%         a,b—— 定解区域[0,a]×[0,b]的上界
%         h——x 和 y 方向的步长
%         ep—— 计算精度
%         Nm—— 迭代次数上限
% 输出:U—— 解矩阵
n = fix(a/h) + 1; m = fix(b/h) + 1;      % x 和 y 方向的网格点数,网格点总数为 n×m
ave = 0;
for i = 1:n
    ave = ave + feval(dfun, (i - 1) * h) + feval(ufun, (i - 1) * h);
end
for k = 1:m
    ave = ave + feval(lfun, (k - 1) * h) + feval(rfun, (k - 1) * h);
end
ave = ave/(2 * (n + m));        % 边界网格点处函数值的平均值
U = ave * ones(n, m);           % 赋迭代初值
U(1, 1:m) = feval(lfun, 0:h:(m - 1) * h)';        % 赋左边界值
U(n, 1:m) = feval(rfun, 0:h:(m - 1) * h)';        % 赋右边界值
U(1:n, 1) = feval(dfun, 0:h:(n - 1) * h)';        % 赋下边界值
U(1:n, m) = feval(ufun, 0:h:(n - 1) * h)';        % 赋上边界值
w = 4/(2 + sqrt(4 - (cos(pi/(n - 1)) + cos(pi/(m - 1)))^2));        % 计算松弛因子
err = 1;
no = 0;
while ((err > ep) & (no < Nm))
    err = 0;
    for j = 2:(m - 1)
        for i = 2:(n - 1)
            re = (U(i,j + 1) + U(i,j - 1) + U(i + 1,j) + U(i - 1,j) - 4 * U(i,j))/4;
            U(i,j) = U(i,j) + w * re;
            if (err <= abs(re))
                err = abs(re);
            end
        end
    end
    no = no + 1;
end
U = flipud(U');        % 按正常坐标方向(y 坐标方向向上)输出结果数据
%D = U';               % 按 x 坐标方向向右,y 坐标方向向下输出数据
```

例 8.5 应用程序(8.3)求解下列定解问题:

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0 \\ u(x,0) = 0, u(x,b) = \sin \dfrac{3\pi x}{a} \cos \dfrac{\pi x}{a} \quad (0 \leqslant x \leqslant a, 0 \leqslant y < b) \\ u(0,y) = 0,\ u(a,y) = \sin \dfrac{3\pi y}{b} \end{cases}$$

其中，$a=3, b=2$。

解　在 MATLAB 命令窗口中输入：

```
>> lfun = inline('0','x');                                  % 定义左边界条件函数
>> rfun = inline('sin(3 * pi * x/2)','x');                  % 定义右边界条件函数
>> dfun = inline('0','x');                                  % 定义下边界条件函数
>> ufun = inline('sin(3 * pi * x/3). * cos(pi * x/3)','x'); % 定义上边界条件函数
>> U = DifLapl(dfun, ufun, lfun, rfun, 3, 2, 0.1, 1e-4, 200);   % 调用程序计算
>> x = 0:0.1:3;      % 绘图时的 x 坐标向量
>> y = 0:0.1:2;      % 绘图时的 y 坐标向量
>> surf(x,y,U)       % 绘制曲面图
```

输出结果如图 8.7 所示。

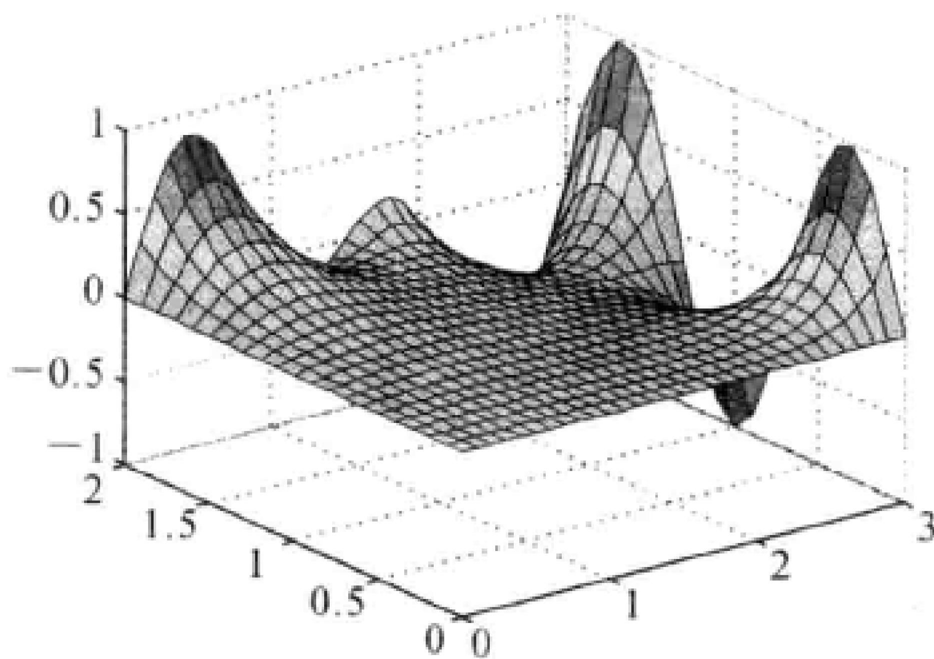


图　8.7

8.6　物理学中的应用举例

8.6.1　扩散现象研究

介质 1 所在区间$[0,a]$的左端有恒定浓度的介质 2，介质 1 的右端处介质 2 的浓度保持为 0，从 $t=0$ 时刻开始介质 2 从左端向介质 1 内扩散，设介质 2 在介质 1 内的扩散系数为 D，分析在介质 1 内介质 2 浓度随时间的变化。

该过程对应的初边值问题是

$$\left.\begin{array}{ll} \dfrac{\partial c}{\partial t} = D \dfrac{\partial^2 c}{\partial x^2} & (0 < x < a, t > 0) \\ c(0,t) = c_0,\quad c(a,t) = 0 & (t > 0) \\ c(x,0) = 0 & (0 < x < a) \end{array}\right\} \tag{8.73}$$

式中，c 是介质 1 内介质 2 的浓度，c_0 是介质 1 左端处介质 2 的浓度。取 $a=1$，$D=0.1$，$c_0=1$，空间步长 $h=0.05$，时间步长 $\tau=0.001$，应用程序(8.1) 进行数值计算，分析扩散过程。

(1)
```
>> fun = inline('0','x');                          % 建立初始条件函数
>> U = DifParab(fun, 1, 0, 1, 1, 0.14, 20, 140);   % 调用程序(8.1) 计算
```

(2) 绘制不同时刻介质 1 内介质 2 的浓度分布。
```
>> x = 0:0.05:1;                  % 建立横坐标向量
>> plot(x,U(11,:),'-k')           % 绘制 t11 = 0.001×(11-1) = 0.01 时刻的浓度分布曲线
>> plot(x,U(51,:),'-k')           % 绘制 t51 = 0.001×(51-1) = 0.05 时刻的浓度分布曲线
>> meshz(U)% 绘制浓度分布三维示意图
```

结果如图 8.8 所示。

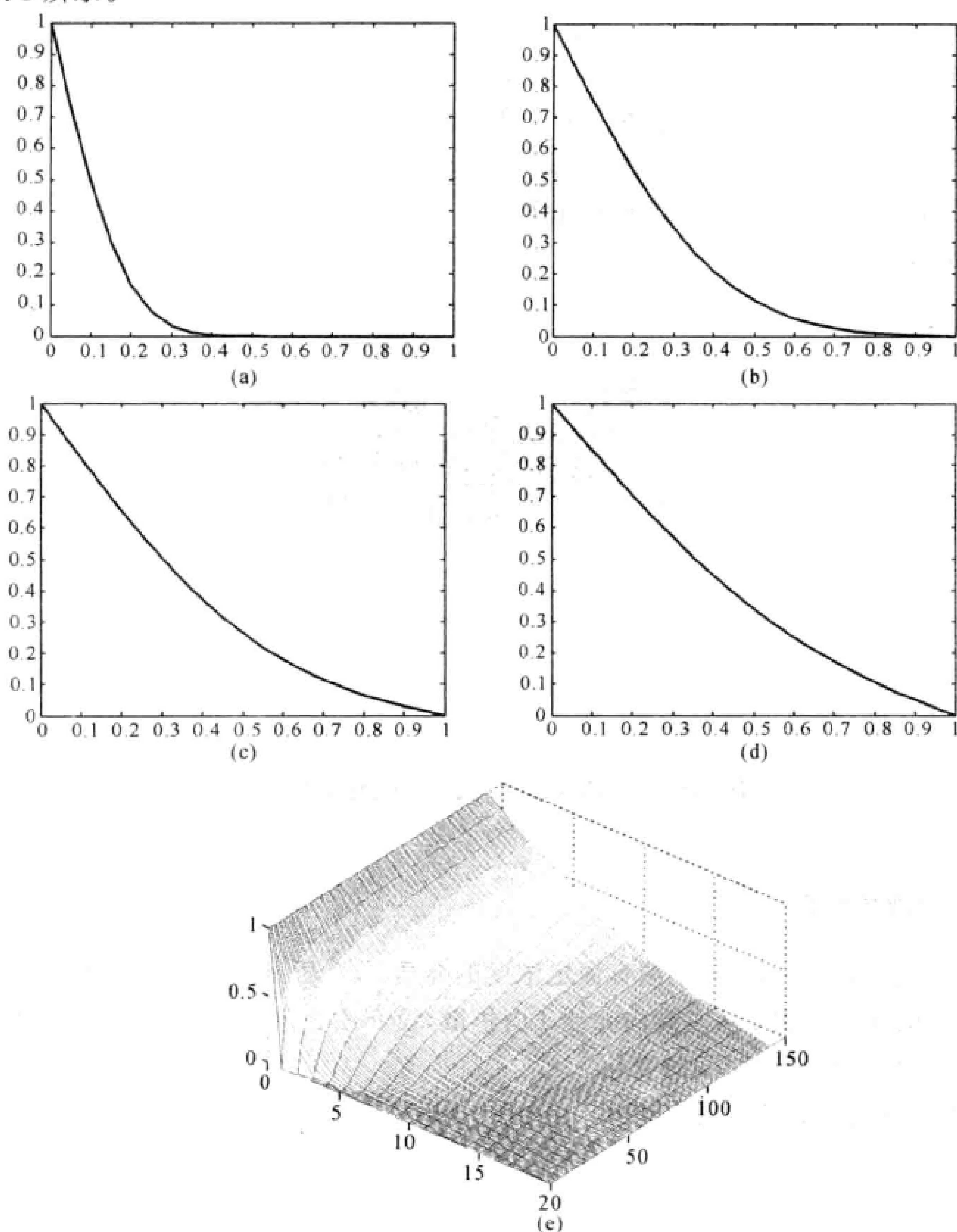


图 8.8

(a)$t=0.01$； (b)$t=0.05$； (c)$t=0.10$； (d)$t=0.14$

从图 8.8 中可以发现：在开始阶段，介质 1 内的介质 2 浓度变化较大，说明介质 2 在介质 1 内的积累较快，因为从左端进入介质 1 的介质 2 几乎全部留在介质 1 内；在 $t=0.05\sim0.10$ 之间，介质 1 内的介质 2 浓度变化缓慢，说明介质 2 在介质 1 内的积累较慢，这是由于从左端进入介质 1 的介质 2 大部分从右端流出；在 $t>0.14$ 后，介质 1 内的介质 2 浓度基本不变，说明介质 2 在介质 1 内的积累变化不明显，原因是从左端进入介质 1 的介质 2 几乎全部从右端流出，因此从宏观上看介质 2 在介质 1 的扩散似乎结束。

8.6.2　平行板电容器内电势的计算

如图 8.9 所示，一平行板电容器的两个极板沿 z 方向为无限长，沿 x 方向的宽度为 10，沿 y 方向的间距为 4，上面的板为电容器正极板，其电势为 +1，下面的板为电容器负极板，其电势为 −1，用差分方法分析两极板之间电势的分布。

两极板之间的空间不存在电荷，电势满足拉普拉斯方程。根据问题的特点，在平行于 xy 坐标面的各平面上电势分布相同，所以可以把问题简化到二维，只研究 xy 坐标面上的电势分布，如图 8.10 所示。由于前面介绍的差分方法只能计算矩形区域的问题，而该问题的区域不是闭合的矩形区域，所以把电容器极板间的空间适当扩大，拓展为如图 8.10 所示的网格区域，把电容器两极板之间划分为 10×4 个网格，左右向外拓展一个网络的宽度，把定解问题的区域扩大为 12×4 个网格。网格节点沿 x 和 y 方向的编号标在图中，共有 13×5 个网格节点。

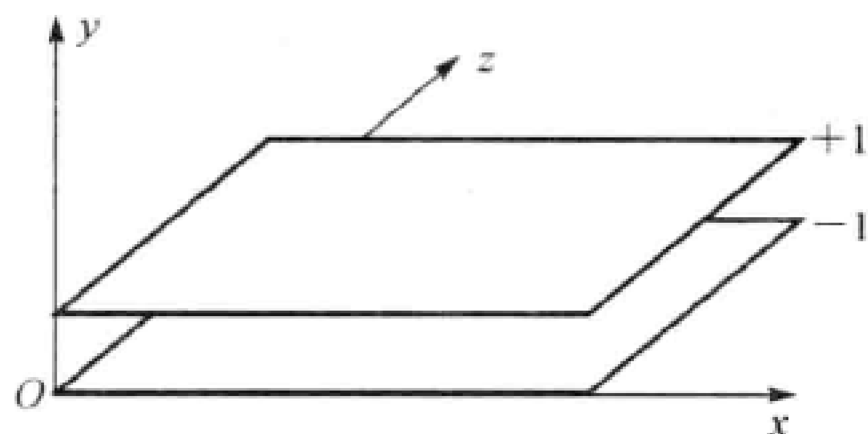


图　8.9

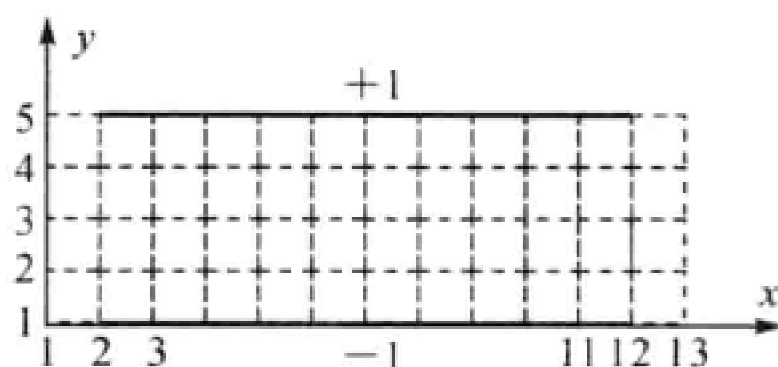


图　8.10

用双重迭代法处理该定解问题，一是左右边界网格节点上的电势值用迭代法确定，二是内部网格节点上的电势值用差分法迭代计算。

边界网格节点上的电势值分两类用不同方法确定：

(1) 节点(2,1)，(3,1)，…，(12,1) 处的电势值始终等于 −1，节点(2,5)，(3,5)，…，(12,5) 处的电势值始终等于 1，节点(1,3)，(13,3) 处的电势值始终等于 0。

(2) 第一次迭代计算内部节点处电势值时，左边界节点(1,1) 处的电势值取 −0.5，节点(1,2) 处的电势值取 −0.25，节点(1,4) 处的电势值取 0.25，节点(1,5) 处的电势值取 0.5。右边界节点处的电势值根据对称性确定。

(3) 以后各次计算内部节点处电势值时，左边界节点(1,1) 处的电势值取上次计算出的节点(2,1) 和(1,2) 处电势值的平均值，节点(1,2) 处的电势值取上次计算出的节点(1,1)，(2,2) 和(1,3) 处电势值的平均值。左、右两端其他边界节点处的电势值可根据节点(1,1) 和(1,2) 处的电势值结合对称性确定。

内部节点处电势值的计算应用程序(8.3) 进行，经过 11 次的迭代计算，在保留到小数点后四位的情况下，全部 13×5 个节点处的电势值达到稳定，不再变化，迭代计算随即结束。11 次

迭代计算过程中，边界节点(1,1)和(1,2)处电势值列于表8.5，各节点处电势$u_{i,k}$的最终计算结果列于表8.6(右半边节点处的电势值根据对称性可得)。电势分布的曲面图与等势线图分别如图8.11和图8.12所示。

表 8.5

迭代次数	1	2	3	4	5	6
$u_{1,1}$	−0.500 0	−0.625 0	−0.655 0	−0.679 1	−0.686 2	−0.690 8
$u_{1,2}$	−0.250 0	−0.311 0	−0.358 1	−0.372 3	−0.381 6	−0.384 8
迭代次数	7	8	9	10	11	
$u_{1,1}$	−0.692 4	−0.693 3	−0.693 7	−0.693 9	−0.694 0	
$u_{1,2}$	−0.386 6	−0.387 3	−0.387 7	−0.387 9	−0.388 0	

表 8.6

k \ i	1	2	3	4	5	6	7
5	0.694 0	1.000 0	1.000 0	1.000 0	1.000 0	1.000 0	1.000 0
4	0.388 0	0.470 0	0.492 0	0.497 8	0.499 4	0.499 8	0.499 9
3	0.000 0	0.000 0	0.000 0	0.000 0	0.000 0	0.000 0	0.000 0
2	−0.388 0	−0.470 0	−0.492 0	−0.497 8	−0.499 4	−0.499 8	−0.499 9
1	−0.694 0	−1.000 0	−1.000 0	−1.000 0	−1.000 0	−1.000 0	−1.000 0

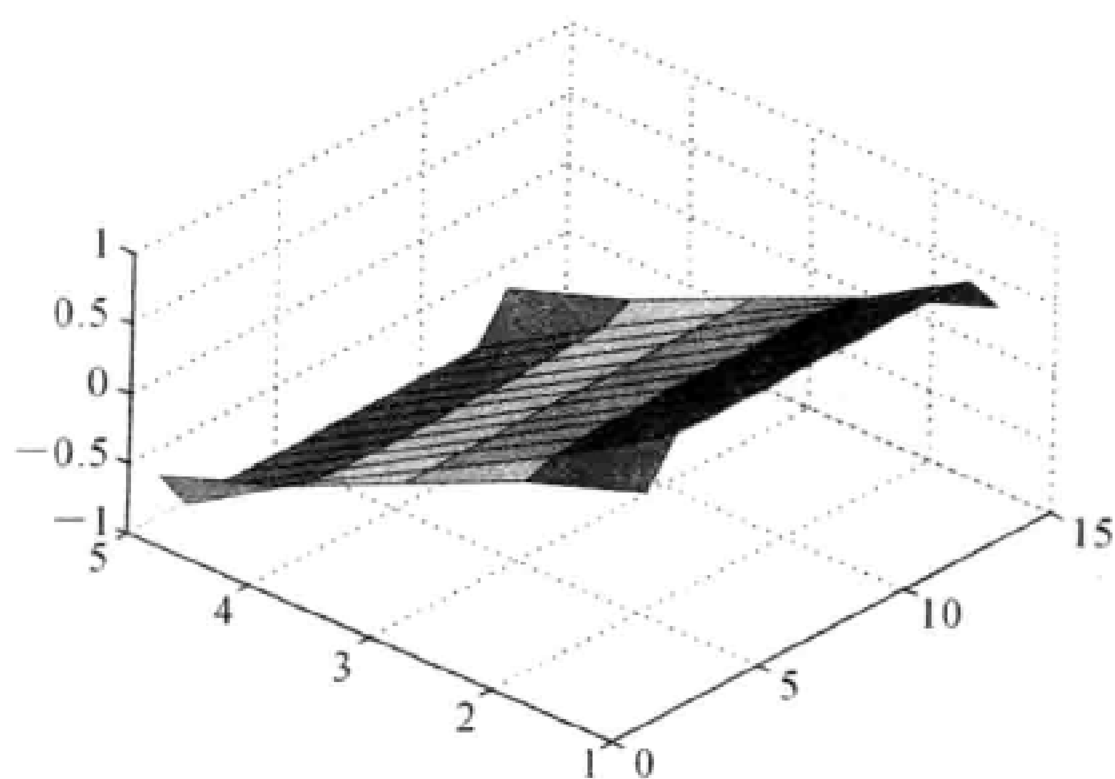


图 8.11

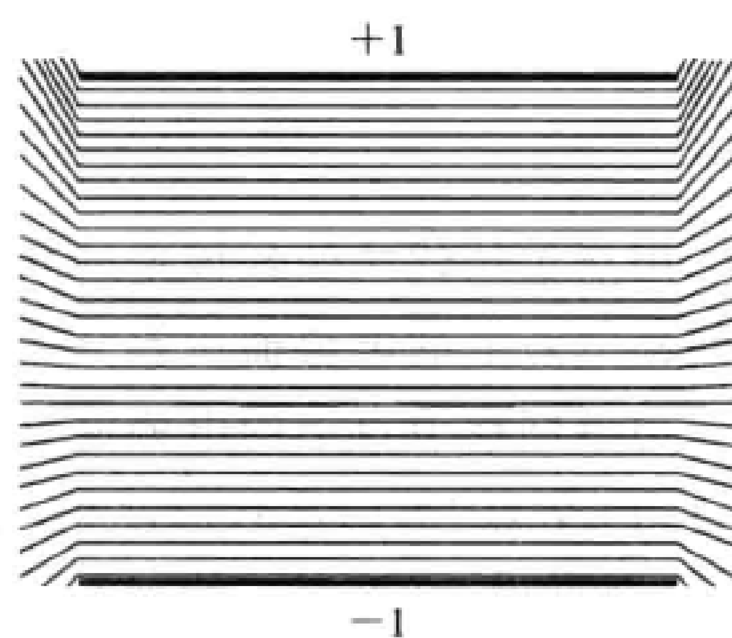


图 8.12

习　　题

1. 试求热传导方程定解问题的数值解：

$$\begin{cases}\dfrac{\partial u}{\partial t}=\dfrac{\partial^2 u}{\partial x^2} \\ u(0,t)=u(1,t)=0 \\ u(x,0)=4x-4x^2\end{cases} \quad (0<x<1,0<t<0.2)$$

取步长 $h=0.2, \tau=0.02$。

2. 两端固定的均匀弦线自由振动定解问题是

$$
\begin{cases}
\dfrac{\partial^2 u}{\partial t^2}=\dfrac{\partial^2 u}{\partial x^2} \\
u(0,t)=u(1,t)=0 \\
u(x,0)=f(x), \quad \dfrac{\partial u}{\partial t}(x,0)=0
\end{cases}
$$

式中

$$
f(x)=\begin{cases}
\sin(7\pi x), & \dfrac{3}{7} \leqslant x \leqslant \dfrac{4}{7} \\
0, & x<\dfrac{3}{7} \text{ 或 } x>\dfrac{4}{7}
\end{cases}
$$

用差分法计算定解问题的数值解，并分析弦线上波传播的特点。

3. 用差分法解拉普拉斯方程 $\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}=0$，计算三个内部节点处的函数值，网格划分及边界条件如图 8.13 所示。

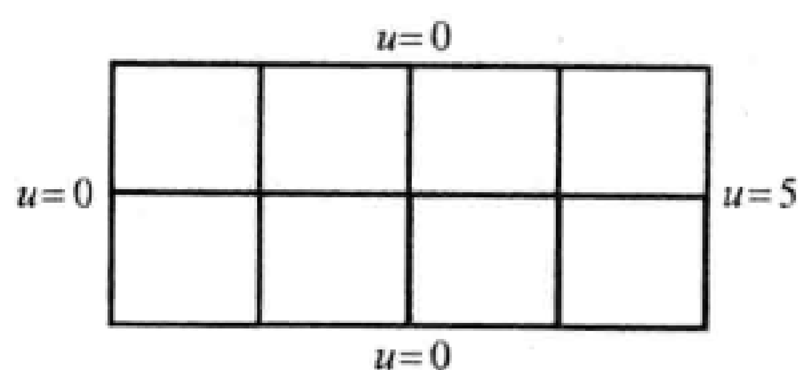


图　8.13

4. 二维温度场的定解问题是

$$
\begin{cases}
\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}=0 \\
u(0,y)=0, \quad u(50,y)=5 \\
u(x,0)=0, \quad u(x,50)=0
\end{cases}
$$

取步长 $h=0.1$，用差分法求解区域内的温度分布。

第 9 章　蒙特卡罗方法简介

蒙特卡罗(Monte Carlo)方法简称为MC方法,也称为统计模拟方法或随机抽样方法,它利用随机数的统计特性进行模拟和计算。蒙特卡罗方法既可以用于随机过程统计性质的研究,也可以用于数值模拟,还可以用于数值计算,因此它广泛应用于粒子输运过程、粒子反应与探测过程、统计物理计算、量子力学计算、多重积分计算、解线性(非线性)方程(组)及微分方程的边值问题等。

蒙特卡罗方法由诺伊曼(John von Neumann)、乌拉姆(Stanislaw Ulam)和梅特罗波利斯(Nicholas Metropolis)于 20 世纪 40 年代首先提出,他们三位是第二次世界大战期间美国"曼哈顿计划"的成员,诺伊曼用驰名的赌城——摩纳哥的蒙特卡罗来命名这种方法。其实在此之前,蒙特卡罗方法就已经存在。1777 年法国数学家布丰(Georges Louis Leclere de Buffon)提出了用投针实验计算圆周率的方法,这被认为是蒙特卡罗方法的起源。

MC 方法大致可以分成两类。一类是求解的问题本身具有随机性。借助计算机的运算能力,可以直接模拟这种随机过程。另一类是求解的问题可以转化为某种随机分布的特征数(比如随机事件的概率或者随机变量的期望值)。通过随机抽样方法,以随机事件的频率估计概率,或者以抽样的数字特征估算随机变量的数字特征。

使用 MC 方法应该注意两点:①要生成具有特定分布的随机数;②要进行误差估计。由于 MC 方法是按概率分布进行抽样计算的,结果必然存在误差。在没有其他方法可用或其他方法很难采用的情况下,MC 方法的计算结果也是有价值的。

9.1　随机变量、概率密度与分布函数

随机变量是指变量的值无法事先确定,但是变量在任意区间取值的概率是知道的。连续型随机变量取值的分布用概率密度函数和分布函数描述。

定义(9.1)　对于连续型随机变量 x,如果它在区间 $[a,b]$ 内取值的概率为

$$P(a \leqslant x \leqslant b) = \int_a^b f(x)\,\mathrm{d}x \tag{9.1}$$

则称函数 $f(x)$ 是随机变量 x 的概率密度函数。

定义(9.2)　概率密度函数为 $f(x)$ 的随机变量 x 的分布函数定义为

$$F(x) = \int_{-\infty}^{x} f(\xi)\,\mathrm{d}\xi \tag{9.2}$$

概率密度函数 $f(x)$ 和分布函数 $F(x)$ 的关系是

$$f(x) = \frac{\mathrm{d}F(x)}{\mathrm{d}x} \tag{9.3}$$

由于 $P(-\infty < x < \infty) = 1$,所以概率密度函数 $f(x)$ 必须满足归一化条件

$$\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x = 1 \tag{9.4}$$

同时，随机变量 x 的值在区间 $[a,b]$ 内的概率是

$$P(a \leqslant x \leqslant b) = \int_a^b f(\xi)\,\mathrm{d}\xi = F(b) - F(a) \tag{9.5}$$

定义(9.3)　若随机变量 x 的概率密度函数是

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{9.6}$$

其中，μ 是任意常数，σ 是大于零的常数。则称随机变量 x 服从参数 μ,σ 的正态分布，记作 $x \sim N(\mu,\sigma^2)$。它的分布函数为

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left[-\frac{(\xi-\mu)^2}{2\sigma^2}\right] \mathrm{d}\xi \tag{9.7}$$

当 $\mu=0,\sigma=1$ 时，称为标准正态分布，记为 $N(0,1)$。正态分布 $N(\mu,\sigma^2)$ 可以通过线性变换转化为标准正态分布，如若 $X \sim N(\mu,\sigma^2)$，令 $Y=(X-\mu)/\sigma$，则 $Y \sim N(0,1)$。

$N(0,1)$ 的曲线如图 9.1 所示。对于标准正态分布，有

$$F(-Y) = 1 - F(Y)$$
$$P(|Y| \leqslant 1) = 2F(1) - 1 = 0.682\,6$$
$$P(|Y| \leqslant 2) = 2F(2) - 1 = 0.954\,4$$
$$P(|Y| \leqslant 3) = 2F(3) - 1 = 0.997\,4$$

所以 Y 的取值几乎全部集中在区间 $[-3,3]$，这称为 3σ 原则，也称为三倍标准差原则。

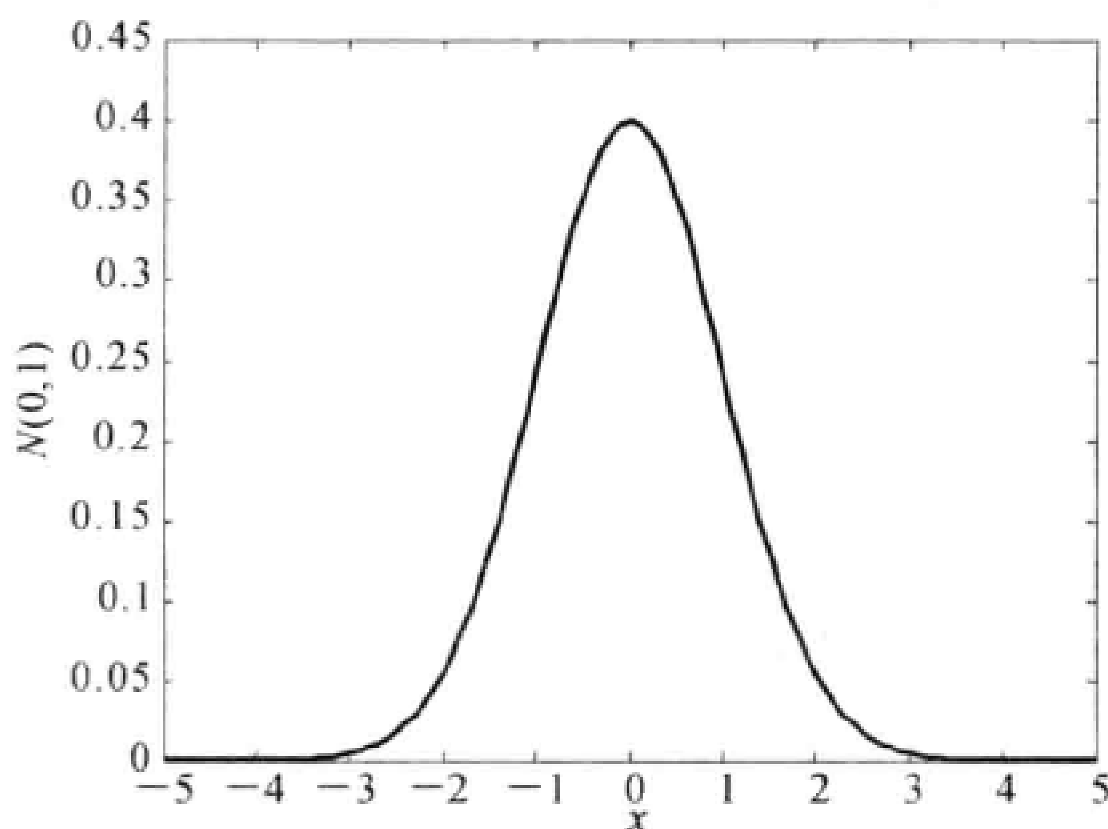


图　9.1

定义(9.4)　随机变量 x 的函数 $g(x)$ 的数学期望值定义为它的平均值，即

$$E\{g\} = \int_{-\infty}^{\infty} g(x)\,\mathrm{d}F(x) = \int_{-\infty}^{\infty} g(x) f(x)\,\mathrm{d}x \tag{9.8}$$

定义(9.5)　随机变量 x 的函数 $g(x)$ 的方差定义为

$$D\{g\} = E\{(g-E\{g\})^2\} = \int_{-\infty}^{\infty} [g-E\{g\}]^2 \mathrm{d}F(x) = \int_{-\infty}^{\infty} [g-E\{g\}]^2 f(x)\,\mathrm{d}x \tag{9.9}$$

方差的平方根称为标准误差，也称为均方根误差或均方差。式(9.9)还能表示为

$$D\{g\}=\int_{-\infty}^{\infty} g^2 f(x)\,\mathrm{d}x-(E\{g\})^2=E\{g^2\}-(E\{g\})^2 \tag{9.10}$$

式中，$E\{g^2\}=\int_{-\infty}^{\infty} g^2 f(x)\,\mathrm{d}x$，是函数 $g(x)$ 的均方值。

服从正态分布 $N(\mu,\sigma^2)$ 的随机变量 x 的数学期望、方差和均方差分别是 μ,σ^2 和 σ。在区间 $[a,b]$ 内满足均匀分布 $f(x)=1/(b-a)$ 的随机变量 x 的数学期望、方差和均方差分别是 $(b+a)/2$，$(b-a)^2/12$ 和 $(b-a)/2\sqrt{3}$。

9.2 随机数的产生

运用 MC 方法计算时首先要产生随机数，基本随机数和具有特殊分布随机数的产生方法不同。

9.2.1 基本随机数的产生与检验

基本随机数是在区间 $[0,1]$ 内均匀分布的随机数，它以相同的概率在区间 $[0,1]$ 内任意取值，具有充分的随机性和均匀性。真正的随机数除了具有统计规律外无任何其他规律可循，因此根本无法给出有限个随机数构成的序列。计算机上使用的随机数是按照一定算法产生的似乎随机的数，称为伪随机数。伪随机数有一定的规律和周期性，但只要周期足够长，那么这种随机数也是可用的。计算机上产生伪随机数的方法很多，通常直接产生在 $[0,1]$ 内等概率分布的随机数，例如 MATLAB 指令的 rand 就产生 $[0,1]$ 内的一维或多维均匀分布的随机数。等概率分布随机数的产生方法有线性同余法（乘加同余法）、组合发生器法、混沌法及人工方法，这些方法可以单独使用，也可以结合使用。这里只介绍线性同余法。

线性同余法产生伪随机数序列 $r_0,r_1,\cdots,r_i,\cdots$ 的递推公式是

$$\left.\begin{aligned} x_{i+1}&=(\lambda x_i+C)\bmod(M)\\ r_{i+1}&=\frac{x_{i+1}}{M}\end{aligned}\quad (i=0,1,2,\cdots)\right\} \tag{9.11}$$

其中，$x=A\bmod(M)$ 表示 x 等于 A 除以 M 的余数。M,λ,C,x_0 都是正整数，M 是模，λ 是乘子，C 为增量，x_0 是初始值或种子。显然 $r_i\in[0,1]$。一般要求 M 要尽可能大。若 $C=0$，则称乘同余法。

取 $x_0=41$，$M=222$，$\lambda=5$，$C=0$ 产生的前 20 个伪随机数列于表 9.1 中，它们的平均值等于 0.510 8，方差等于 0.060 7，具有较好的统计性质。

表 9.1

i	1	2	3	4	5	6	7	8	9	10
r_i	0.184 7	0.923 4	0.617 1	0.085 6	0.427 9	0.139 6	0.698 2	0.491 0	0.455 0	0.274 8
i	11	12	13	14	15	16	17	18	19	20
r_i	0.373 9	0.869 4	0.346 8	0.734 2	0.671 2	0.355 9	0.779 3	0.896 4	0.482 0	0.409 9

产生的随机数应该具有充分长的周期和真正随机数所具有的统计性质。因此必须对随机数进行检验，检验主要包括参数检验（亦称矩检验，检验随机数分布参数的实际值与理论值的

差异是否显著）、均匀性检验（亦称频率检验，检验基本随机数在区间[0,1] 内分布是否均匀）、随机性检验（亦称连检验，检验随机数序列的随机性是否良好）和独立性检验（检验随机数之间是否独立）几个方面。这里只给出参数检验方法。

（1）观测值：取随机数的 n 个抽样值 $r_1, r_2, \cdots, r_n$，它们的平均值、均方值和方差分别是

$$\langle r_n \rangle = \frac{1}{n}\sum_{i=1}^{n} r_i, \quad \langle r_n^2 \rangle = \frac{1}{n}\sum_{i=1}^{n} r_i^2, \quad D_n = \frac{1}{n}\sum_{i=1}^{n}\left(r_i - \frac{1}{2}\right)^2 = \langle r_n^2 \rangle - \langle r_n \rangle + \frac{1}{4} \tag{9.12}$$

（2）理论值：若随机数 $r_0, r_1, \cdots, r_i, \cdots$ 是均匀分布在区间[0,1]内的，则其平均值、均方值和方差的理论值分别是

$$\langle r \rangle = \frac{1}{2}, \quad \langle r^2 \rangle = \frac{1}{3}, \quad D = \frac{1}{12} \tag{9.13}$$

（3）概率误差：用 $\langle x_n \rangle$ 表示上述三个观测值（$\langle r_n \rangle$, $\langle r_n^2 \rangle$ 和 D_n）中的一个，它是随机的，并且与抽样值的个数 n 有关。$\langle x_n \rangle$ 与理论值 $\langle x \rangle$（$\langle r \rangle$, $\langle r^2 \rangle$, D 中的一个）的差异 $|\langle x_n \rangle - \langle x \rangle|$ 只能是在一定概率保证下的差异。若以不低于 $(1-\alpha)$ 的概率保证 $\langle x_n \rangle$ 与 $\langle x \rangle$ 的相对误差小于 ε，即

$$P\left(\frac{|\langle x_n \rangle - \langle x \rangle|}{|\langle x \rangle|} \leqslant \varepsilon\right) = 1 - \alpha \tag{9.14}$$

其中，α 称为可信度，$(1-\alpha)$ 称为可信水平。

定理(9.1)（大数定理）　设随机变量 $x_1, x_2, \cdots, x_n, \cdots$ 相互独立且具有相同的数学期望与方差，即 $E\{x_k\} = \mu, D\{x_k\} = \sigma^2 (k = 1, 2, \cdots)$，作前 n 个随机变量的算术平均值 $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$，则对于任意的 $\varepsilon > 0$，有

$$\lim_{n\to\infty} P\{|\bar{x} - \mu| < \varepsilon\} = \lim_{n\to\infty} P\left\{\left|\frac{1}{n}\sum_{i=1}^{n} x_i - \mu\right| < \varepsilon\right\} = 1 \tag{9.15}$$

定理(9.2)（中心极限定理）　设随机变量 $x_1, x_2, \cdots, x_n, \cdots$ 相互独立且具有相同的数学期望与方差，即 $E\{x_k\} = \mu, D\{x_k\} = \sigma^2 (k = 1, 2, \cdots)$，作前 n 个随机变量的算术平均值 $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$，则对于任意的实数 λ，有

$$\lim_{n\to\infty} P\left\{\frac{\frac{1}{n}\sum_{i=1}^{n} x_i - \mu}{\frac{\sigma}{\sqrt{n}}} < \lambda\right\} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\lambda} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t \tag{9.16}$$

根据中心极限定理，当 n 充分大时，有

$$P\left(\frac{|\langle x_n \rangle - \langle x \rangle|}{|\langle x \rangle|} \leqslant \frac{\tau\sigma}{\sqrt{n}\,|\langle x \rangle|}\right) = \sqrt{\frac{2}{\pi}}\int_0^{\tau} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t \tag{9.17}$$

式中，σ 是标准差，$\sigma^2 = \langle x^2 \rangle - (\langle x \rangle)^2$。比较式(9.14)和式(9.17)，可知

$$\varepsilon = \frac{\tau\sigma}{\sqrt{n}\,|\langle x \rangle|}, \quad 1 - \alpha = \sqrt{\frac{2}{\pi}}\int_0^{\tau} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t \tag{9.18}$$

式(9.17)说明

$$|\langle x_n \rangle - \langle x \rangle| \leqslant \frac{\tau\sigma}{\sqrt{n}} \tag{9.19}$$

成立的概率是 $(1-\alpha)=\sqrt{\frac{2}{\pi}}\int_0^{\tau}\exp\left(-\frac{t^2}{2}\right)\mathrm{d}t$。当 $\tau=1,2,3$ 时，分别有 $(1-\alpha)=0.6826$，0.9544，0.9974。特别称 $\alpha=0.5$ 时的误差 $0.6745\sigma/\sqrt{n}$ 为概率误差。

(4) 用作检验的统计量：在进行随机数检验时，一般取可信水平为 0.95(即可信度 $\alpha=0.05$)，相应的 $\tau=1.96$。将检验的随机数 $r_1,r_2,\cdots,r_n$ 计算出的 $\varepsilon=|\langle x_n\rangle-\langle x\rangle|/|\langle x\rangle|$、观测值的 n 和理论值 $\langle x\rangle$ 代入式(9.18)第一式，有

$$\tau=\frac{\varepsilon\sqrt{n}\ |\langle x\rangle|}{\sigma}\tag{9.20}$$

若得到的 $\tau\leqslant 1.96$，则通过参数检验。对于前面的三个观测值，能够计算出 $\sigma(r)$，$\sigma(r^2)$，$\sigma(D)$ 分别等于 $1/(2\sqrt{3})$，$2/(3\sqrt{5})$ 和 $1/(6\sqrt{5})$。

9.2.2 任意分布随机数的产生

通过随机抽样，就可以从区间[0,1]内均匀分布的基本随机数 $r_0,r_1,\cdots,r_i,\cdots$ 产生具有不同分布函数的随机数。

1. 连续分布的随机抽样

设连续随机变量的分布函数是 $F(x)$，显然 $F(x)$ 是区间[0,1]中的随机数。若 r_i 是区间[0,1]中均匀分布的随机数，令

$$F(x)=r_i$$

得到

$$x_i=F^{-1}(r_i)\tag{9.21}$$

则 $x_i(i=0,1,\cdots)$ 就是具有概率密度函数 $f(x)$ 的随机数。这里 F^{-1} 表示 F 的反函数。

例 9.1　指数分布的随机抽样。

解　指数分布的概率密度函数为

$$f(x)=\lambda\exp(-\lambda x)\quad(0\leqslant x\leqslant\infty)$$

对应的分布函数为

$$F(x)=1-\exp(-\lambda x)$$

令

$$1-\exp(-\lambda x)=r_i$$

得

$$x_i=-\frac{1}{\lambda}\ln(1-r_i)$$

由于 $(1-r_i)$ 与 r_i 的分布相同，上式又能改写为

$$x_i=-\frac{1}{\lambda}\ln r_i$$

2. 舍选抽样法

定理(9.3)　若 x 是区间 $[a,b]$ 上均匀分布的随机数，则利用条件 $f(x)/M\geqslant r$ 选出的 x 将是区间 $[a,b]$ 上概率密度函数为 $f(x)$ 的随机数。其中，r 是基本随机数，M 是 $f(x)$ 的最大值。

如果要产生在[0,1]内按概率密度函数 $f(x)$ 分布的随机数 x，先画一个矩形，宽为 1、高为 M，曲线 $f(x)$ 将包含在这个矩形内，如图 9.2 所示。在矩形内产生均匀的投点，如果这个点

在曲线 $f(x)$ 下边，则保留这个点，反之就舍去。具体方法是，产生[0,1]中均匀分布的两个随机数 s' 和 s''，由它们构成的投点坐标为 $P(r',r'')$，这里

$$r' = s', \quad r'' = Ms'' \tag{9.22}$$

如果 $r'' < f(r')$，则点 P 在曲线 $f(x)$ 之下，随机变量的取值为 $x = r'$，否则就舍弃它，重新选点。可以证明，这样产生的随机数符合要求。

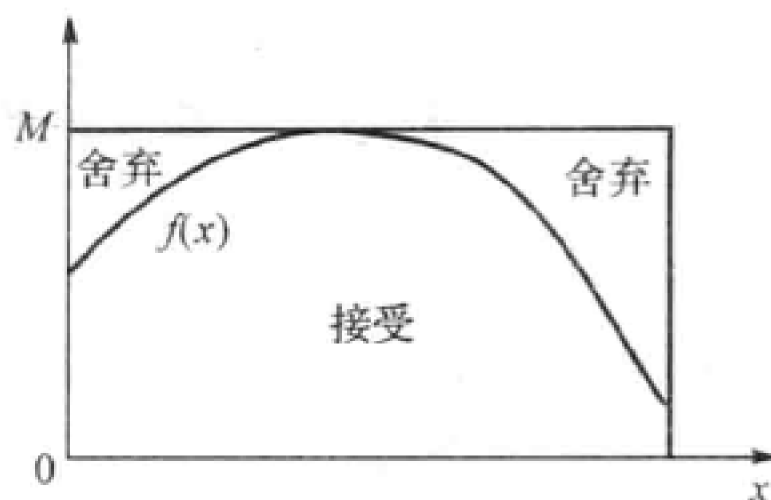


图　9.2

例 9.2　氢原子 1s 态电子云的模拟。这是一个应用舍选抽样法的例子。氢原子基态的电子分布概率密度函数是 $D(r) = (4r^2/a^3)\exp(-2r/a)$，$a = 5.29 \times 10^{-2}$ nm，D 的最大值 $Dr_{\max} = 1.1$，$r_0 = 0.25$ nm 是 D 的收敛点。

解　模拟就是用点在空间的分布密度表示电子在空间分布的概率密度。模拟的方法如下：

(1) 产生一个随机的电子轨道半径 $r_1 = r_0 \text{rand}(1)$，计算出 $Dr = D(r_1)$；

(2) 再产生一个随机的概率判据 $r_2 = Dr_{\max}\text{rand}(1)$，如果 $r_2 > Dr$，则舍弃它，反之计算一个随机角 $\theta = 2\pi\text{rand}(1)$，得到点的坐标 $x = r_1\cos\theta, y = r_1\sin\theta$；

(3) 在点 (x,y) 处画点，返回(1)重复这个过程。

程序(9.1)　模拟氢原子电子云的 MATLAB 程序。

```
N = 300000;  a = 0.529;                      % 画点的总数为 N
r1 = 5 * rand(1,N);                          % 产生随机的电子轨道半径 r1
Dr = 4. * r1.^2./a^3. * exp(-2. * r1./a);    % 计算 r1 处的电子概率密度
M = max(Dr);                                 % 找出 Dr 的最大值
r2 = M * rand(1,N);                          % 产生随机的概率判据 r2
r = r1(find(r2 < Dr));                       % 满足条件(r2 < Dr) 的 r1
n = length(r);                               % 满足条件(r2 < Dr) 的 r1 的数目
Q = 2 * pi * rand(1, n);                     % 产生随机角 θ
[x,y] = pol2cart(Q, r);                      % 计算点的坐标(x,y)
plot(x,y,'k.','markersize',1);               % 画点
axis equal                                   % 图形的纵横坐标等比例
figure                                       % 以下绘制
r = 0:0.01:5;
Dr = 4. * r.^2./a^3. * exp(-2. * r./a);
plot(r, Dr,'k-')
```

程序运行结果如图 9.3($D(r)-r$ 曲线)和图 9.4(电子云模拟结果)所示。

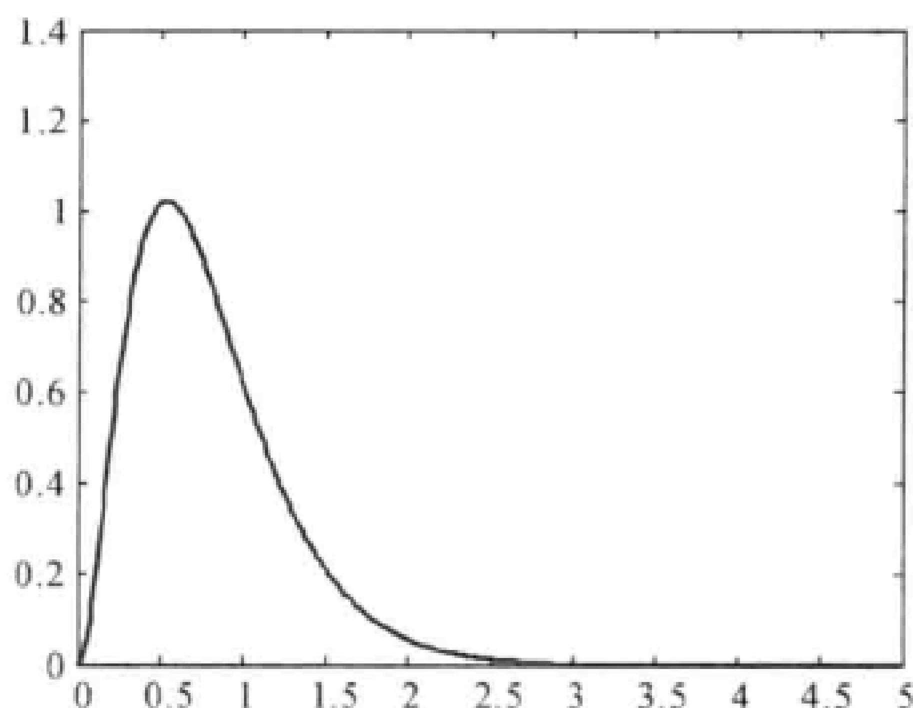


图　9.3

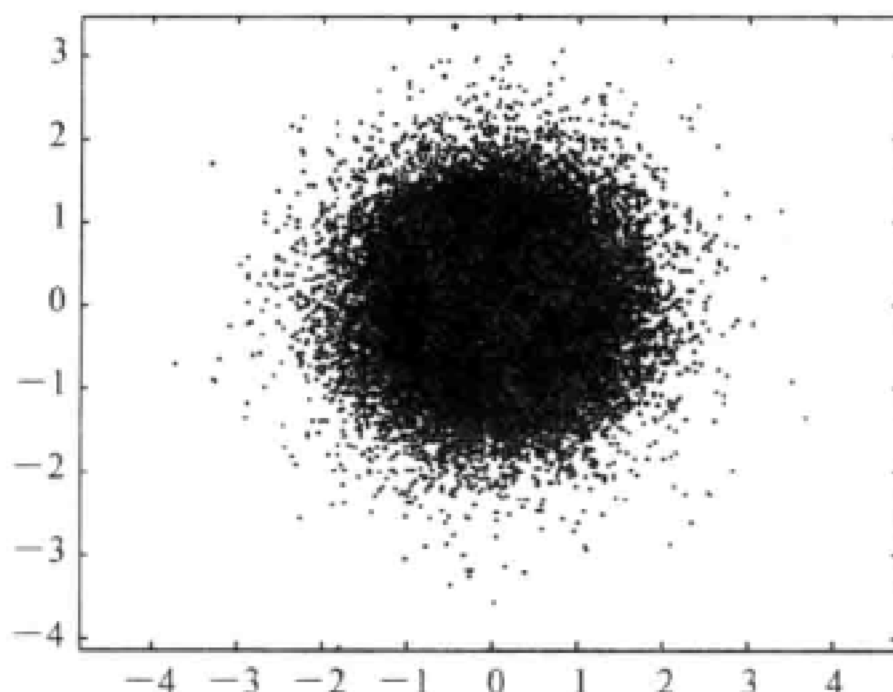


图　9.4

9.3 蒙特卡罗方法在数值分析中的应用

用MC方法求解问题的过程主要有两步：① 建立一个随机变量模型，使问题的解是随机变量的概率或随机变量的期望值等；② 用计算机进行模拟实验，也就是对模型进行随机抽样，产生随机变量，计算随机变量的概率或期望值并作为问题的近似解。

9.3.1 计算定积分的投点法

设函数 $f(x)$ 在区间 $[a,b]$ 内为正，则计算定积分 $I=\int_a^b f(x)\,dx$ 就是计算曲线 $f(x)$ 下的面积，如图9.5所示。为了计算积分，在 xy 平面上作一个包围曲线 $f(x)$ 的矩形，矩形的四个边分别是直线 $x=a$，$x=b$，$y=0$，$y=f_{\max}$，这里 $f_{\max}$ 是函数 $f(x)$ 在区间 $[a,b]$ 内的最大值。然后，随机地向矩形内投掷 N 个点，若落在曲线 $f(x)$ 以下区域的点数为 M，则当 N 足够大时，就有

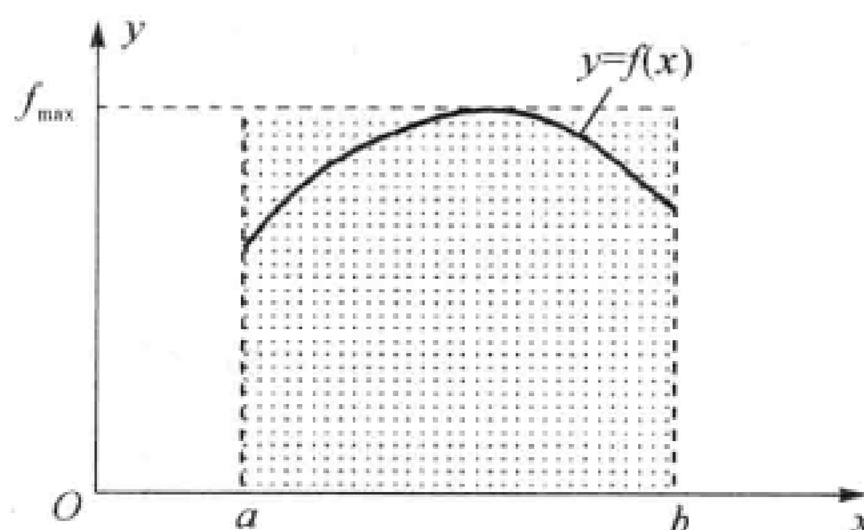


图 9.5

$$\frac{G}{f_{\max}(b-a)}=\frac{\text{曲线下区域的面积}}{\text{矩形的面积}}\approx\frac{\text{落在曲线下区域的点数}}{\text{投掷到矩形内的点数}}=\frac{M}{N}$$

即

$$G\approx f_{\max}(b-a)\frac{M}{N} \tag{9.23}$$

具体计算步骤如下：

(1) 产生两组在区间 $[0,1]$ 内均匀分布的随机数 $r_{11},r_{12},\cdots,r_{1i},\cdots$ 和 $r_{21},r_{22},\cdots,r_{2i},\cdots$；

(2) 随机地向矩形内投掷点，投掷点的坐标是 $p_i(x_i,y_i)$，其中 $\begin{cases}x_i=a+(b-a)r_{1i}\\ y_i=f_{\max}r_{2i}\end{cases}$；

(3) 若点 $p_i(x_i,y_i)$ 在曲线 $f(x)$ 下边，则将该点计入 M，否则不计入 M；

(4) 当向矩形内投掷的点数 N 足够大时，就可以用式(9.23)计算积分的值。

同理，计算二重积分就是计算空间体积 v。这时要构造一个更大的体积为 V 的空间区域，它包围所计算的体积 v，而且易于计算。向区域 V 上均匀投掷 N 个点，数出落在体积 v 内的点的数目 M，当 N 充分大时，有

$$v\approx V\frac{M}{N} \tag{9.24}$$

计算三重积分就是计算“四维”空间区域的体积 v'。这时要构造一个更大的体积为 V' 的空间区域，区域 V' 包围区域 v'，而且便于计算。向区域 V' 内均匀投掷 N 个点，数出落在区域 v' 内的点的数目 M，当 N 充分大时，有

$$v'\approx V'\frac{M}{N} \tag{9.25}$$

例 9.3 计算单位圆的面积。

解 如图9.6所示，随机投掷点的坐标 (x_i,y_i) 满足 $x_i^2+y_i^2\leqslant 1$ 时，点落在圆内。由于对

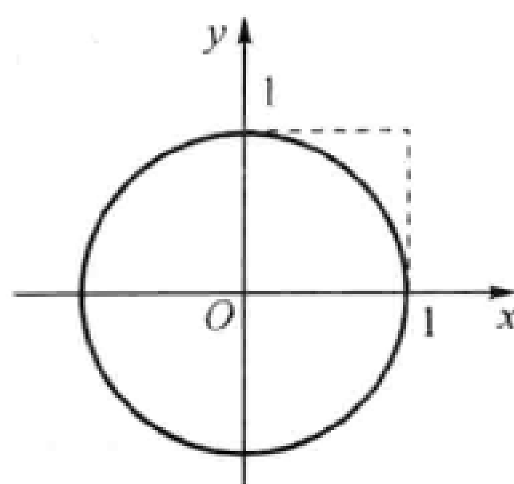


图　9.6

称性，只计算第一象限的面积即可，并且$\frac{S}{4}=\frac{M}{N}$，故单位圆的面积为$S=4M/N$。

程序 (9.2)　用投点法计算单位圆面积（亦即 π 值）的 MATLAB 程序。

```
a = 0; b = 1;          % 积分区间[a, b]的下限和上限
fm = 1;                % 被积函数在区间[a, b]中的最大值
N = 10^5;              % 向矩形内投掷点的总数
r1 = rand(1,N);  r2 = rand(1,N);     % 生成两组在[0, 1]内均匀分布的随机数
x = a + (b - a). * r1;  y = fm. * r2;  % 计算投掷点的坐标
M = 0;                               % 初始化参数 M
for i = 1:N
    if x(i)^2 + y(i)^2 <= 1
        M = M + 1;                   % 若投掷点落在圆内,则 M 值增加 1
    end
end
S = 4 * M/N                          % 计算单位圆的面积
```

N 取不同值的计算结果列入表 9.2 中。

表　9.2

投掷点总数 N	10^4	10^5	10^6	10^7
第 1 次计算结果	3.104 0	3.145 3	3.140 1	3.141 1
第 2 次计算结果	3.150 0	3.144 8	3.141 5	3.141 4
第 3 次计算结果	3.158 0	3.144 2	3.143 9	3.141 6

注：$\pi=3.141\ 592\ 653\ 589\ 79$。

9.3.2　计算定积分的平均值法

数值求积公式为$\int_0^1 f(x)\,dx \approx \sum_{i=1}^{N} A_i f(x_i)$，对于矩形积分公式，$A_i=1/N$，即$\int_0^1 f(x)\,dx \approx \sum_{i=1}^{N} f(x_i)\Big/N$，说明函数 $f(x)$ 在区间 $[0,1]$ 上的积分就等于函数在区间内 N 个点处的平均值。

同理，如果在区间 $[a,b]$ 内等概率地取 N 个点 $x_i\,(i=1,2,\cdots,N)$，则有

$$\int_a^b f(x)\,dx = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i) \tag{9.26}$$

这相当于把 x 看作在区间 $[a,b]$ 内均匀分布的随机数，计算函数 $f(x)$ 的数学期望。这就是 MC 方法的积分公式，它包括两步，一是以一定概率随机选取样点，二是计算平均值，把平均值作为积分值的近似值。积分公式(9.26) 的误差是

$$e=\frac{b-a}{\sqrt{N}}\left[\frac{1}{N}\sum_{i=1}^{N} f_i^2-\left(\frac{1}{N}\sum_{i=1}^{N} f_i\right)^2\right]^{1/2} \tag{9.27}$$

其中，$f_i = f(x_i)$。同样，二重积分和三重积分的 MC 方法分别是

$$I_2 = \frac{S}{N}\sum_{i=1}^{N} f(x_i, y_i) \tag{9.28}$$

$$I_3 = \frac{V}{N}\sum_{i=1}^{N} f(x_i, y_i, z_i) \tag{9.29}$$

式中，S 是 xy 平面上积分域的面积，(x_i, y_i) 是积分域上均匀分布的点的坐标；V 是 xyz 空间内积分域的体积，(x_i, y_i, z_i) 是积分域中均匀分布的点的坐标。

例 9.4 计算积分 $I=\int_0^1 \sqrt{1-x^2}\,dx$。(积分的理论值是 $\pi/4$)。

解 有关参数和函数分别是 $a=0, b=1, f(x)=\sqrt{1-x^2}$。

程序(9.3) 用平均值法计算计算积分 $I=\int_a^b f(x)\,dx$ 的 MATLAB 程序。

```
a = 0; b = 1;           % 积分区间[a, b]的下限和上限
N = 10^5;               % 在区间[a, b]内随机取点的数目
r = rand(1,N);          % 生成在[0, 1]内均匀分布的随机数
x = a+(b-a).*r;         % 计算区间[a, b]内均匀分布的随机点坐标向量
y = sqrt(1-x.^2);       % 计算随机点处函数 f(x) 的值向量
I = mean(y)             % 计算函数值的平均值，并将其作为积分值
```

N 取不同值时的计算结果列入表 9.3 中。

表 9.3

投掷点总数 N	10^4	10^5	10^6	10^7
第 1 次计算结果	0.785 344 3	0.784 563 1	0.785 249 1	0.785 406 9
第 2 次计算结果	0.782 516 9	0.784 704 1	0.785 627 4	0.785 380 4
第 3 次计算结果	0.786 234 6	0.784 939 1	0.785 504 0	0.785 436 3

注：$\pi/4 = 0.785\,398\,163\,397\,45$。

9.3.3 求一元方程的实根

利用逐步逼近方法计算方程 $f(x)=0$ 根的近似值 x_0，并且使 $|f(x_0)|<\varepsilon$，ε 是事先给定的。设方程 $f(x)=0$ 的根在区间 $[a,b]$ 内，计算根近似值的步骤如下：

(1) 在区间 $[a,b]$ 内随机地选择 N 个均匀分布的点 $x_i (i=1,2,\cdots,N)$，计算函数 $f(x)$ 在 x_i 的值 $f(x_i)\ (i=1,2,\cdots,N)$；

(2) 若在 $x_i (i=1,2,\cdots,N)$ 中存在一点 x_m，满足 $|f(x_m)|<\varepsilon$，则方程根的近似值就是 x_m，计算结束，否则进入(3)；

(3) 从 $f(x_i)\ (i=1,2,\cdots,N)$ 中选出 $f(a)\cdot f(x_i)>0$ 且 $|f(x_i)|$ 最小的点 x_a 及 $f(b)\cdot f(x_i)>0$ 且 $|f(x_i)|$ 最小的点 x_b，若 $|f(x_a)|<\varepsilon$ 或 $|f(x_b)|<\varepsilon$，则方程根的近似值为 x_a 或 x_b，否则令 $a=x_a, b=x_b$，从第(1)步重新计算。

例 9.5 求方程 $f(x)=\exp(-x^3)-\tan x+800=0\ (0<x<\pi/2)$ 的根。

解 在 MATLAB 命令窗口定义函数：

```
>> fun = inline('exp(- x^3) - tan(x) + 800', 'x');      % 定义函数 f(x)
```

程序(9.4)　用逐步逼近方法求解方程根的 MATLAB 程序。

```
a = 0; b = pi/2;           % 根所在区间的下限和上限
ep = 1e - 6;               %ε = 10^-6
N = 100;                   % 在区间[a,b]内布置随机点的数目为 100
fa = feval(fun,a); fb = feval(fun,b);        % 计算函数 f(x) 在 a,b 处的值
if fa * fb > 0 error('区间[a,b]内不存在方程的根'); end
while abs(fa) >= ep & abs(fb) >= ep
    x = a + (b - a) * rand(1,N);             % 区间[a,b]内随机点的坐标向量
    for i = 1:N
        fi = feval(fun, x(i));               % 计算函数 f(x) 在 x(i) 处的值
        if abs(fi) < ep
            x0 = x(i); f0 = fi; end          % 当 abs(fi) < ep 时,直接给出根的值和函数值
        if fa * fi > 0 & abs(fi) < abs(fa)
            a = x(i); fa = fi;               % 函数值与 f(a) 同号且绝对值最小的点的坐标 a
        elseif fb * fi > 0 & abs(fi) < abs(fb)
            b = x(i); fb = fi;               % 函数值与 f(b) 同号且绝对值最小的点的坐标 b
        end
    end
end
if abs(fa) < ep
    x0 = a; f0 = fa;
else
    x0 = b; f0 = fb;
end
```

程序运行结果：

```
>> x0 = 1.56954636014902
>> f0 = 1.211268454426318e - 007
```

即方程根的近似值为 1.569 546 360 149 02,此处的函数值等于 1.211×10^{-7},满足计算要求。

9.4　蒙特卡罗方法在数值模拟中的应用

9.4.1　气体自由膨胀过程的模拟

气体的自由膨胀是展示不可逆过程方向性的典型例子。如图 9.7 所示,一个密闭的盒子被隔成 L 和 R 两个体积相等的部分。开始时,L 部分充满气体,而 R 部分为真空。打开两部分之间的隔板后,气体进行自由膨胀,最终气体均匀地充满整个盒子。气体的自由膨胀过程是一个自发过程,相反的过程从来不会自动发生,因此自由膨胀过程是一个不可逆过程。热力学第二定律揭示了孤立系统不可逆过程的方向性。

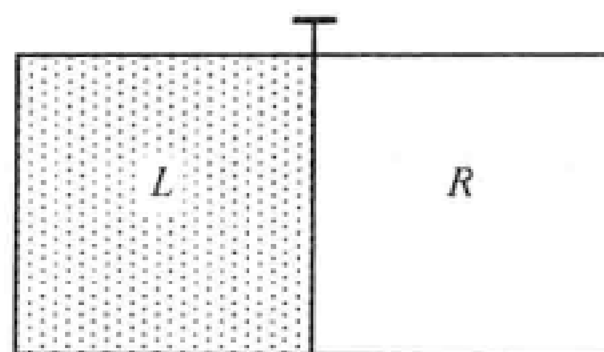


图 9.7

现在对气体的自由膨胀过程进行分析和模拟。设共有 N 个分子，开始时所有分子都在盒子的 L 部分。第一次，随机地选择一个分子，令其从 L 部分进入 R 部分；之后每次，随机地选择一个分子，令其从所在的部分进入另一部分；如此不断进行。

首先对这一过程进行解析分析。令 n 为第 x 次随机选择并移动分子后 R 部分中的分子数，则在下次选择时，选中 R 部分中分子并移动到 L 部分的概率为

$$P_R = \frac{n}{N}$$

而选中 L 部分中分子并移动到 R 部分的概率为

$$P_L = 1 - \frac{n}{N}$$

则每选择并移动一个分子时，R 部分中的分子数的增量是

$$\Delta n_0 = P_L - P_R = 1 - \frac{2n}{N}$$

因此如果连续有 Δx 个分子发生移动，那么 R 部分中的分子数的增量是

$$\Delta n = \left(1 - \frac{2n}{N}\right)\Delta x$$

把 n 和 x 看成连续变量，有

$$\frac{\mathrm{d}n}{\mathrm{d}x} = 1 - \frac{2n}{N} \tag{9.30}$$

结合初始条件 $n(x=0)=0$，解微分方程式(9.30)，得 R 部分中的分子数为

$$n = \frac{N}{2}\left[1 - \exp\left(-\frac{2x}{N}\right)\right] \tag{9.31}$$

L 部分中的分子数为

$$N - n = \frac{N}{2}\left[1 + \exp\left(-\frac{2x}{N}\right)\right] \tag{9.32}$$

显然，当 $x \to \infty$ 时，R 部分和 L 部分中的分子数均趋于 $N/2$，即气体达到平衡状态。

然后对这一过程进行模拟。方法如下：

(1) 用数组 B 的元素表示每一个分子及其所在位置。元素的序号为分子的编号，元素的值代表分子在盒子中的位置(元素值等于 1 时，表示该分子在盒子的 L 部分；元素值等于 -1 时，表示该分子在盒子的 R 部分)。

(2) 每次随机地选择一个分子，令其从所在的部分进入另一部分，就相当于使代表该分子位置的元素值改变符号。

(3) 经过若干次随机选择数组 B 的元素并改变其值的符号后，统计值等于 1 的元素的数目，就等于盒子 L 部分中的分子数目。

程序(9.5)　模拟气体自由膨胀的 MATLAB 程序。

```
N = 2000;          % 总分子数
B = ones(1, N); % 开始时分子均在盒子的 L 部分，数组 B 所有元素都等于 1
for i = 1:10000    % 最多进行 10 000 次随机选择和移动
    k = ceil(rand(1) * N);          % 随机选择第 k 个分子
    B(k) = -B(k);                   % 第 k 个分子所在部分发生变化
    cn = length(find(B == 1));      % 计算盒子 L 部分中的分子数目
    n(i) = cn;                      % 经过 i 次移动后盒子 L 部分中的分子数目
end
plot(n./N,'k.');                    % 绘制 L 部分分子数曲线
xlabel('x'); ylabel('(N-n)/N');
x = 1:10000;
y = N * 0.5 * (1 + exp(-2 * x./N));% 用解析式(9.32)计算盒子 L 部分中的分子数目
figure
plot(x,y./N,'k-');
xlabel('x'); ylabel('(N-n)/N');
```

程序运行结果如图 9.8 和图 9.9 所示。

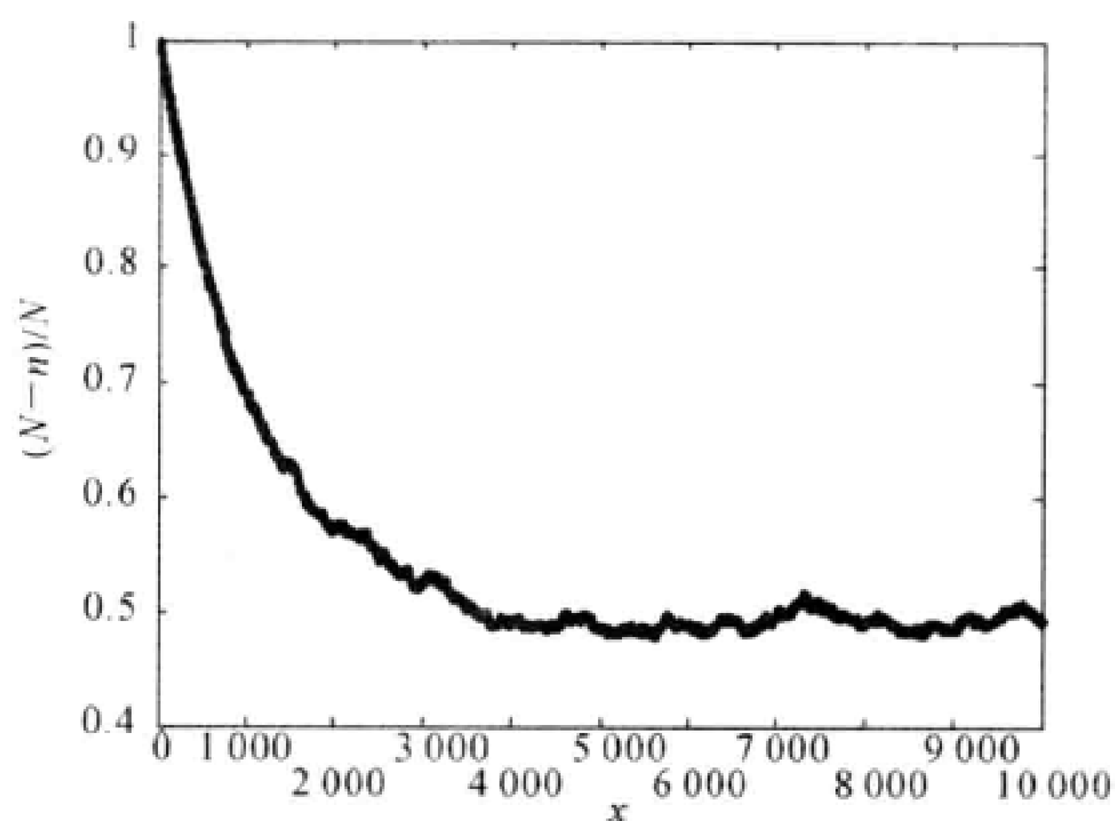


图　9.8

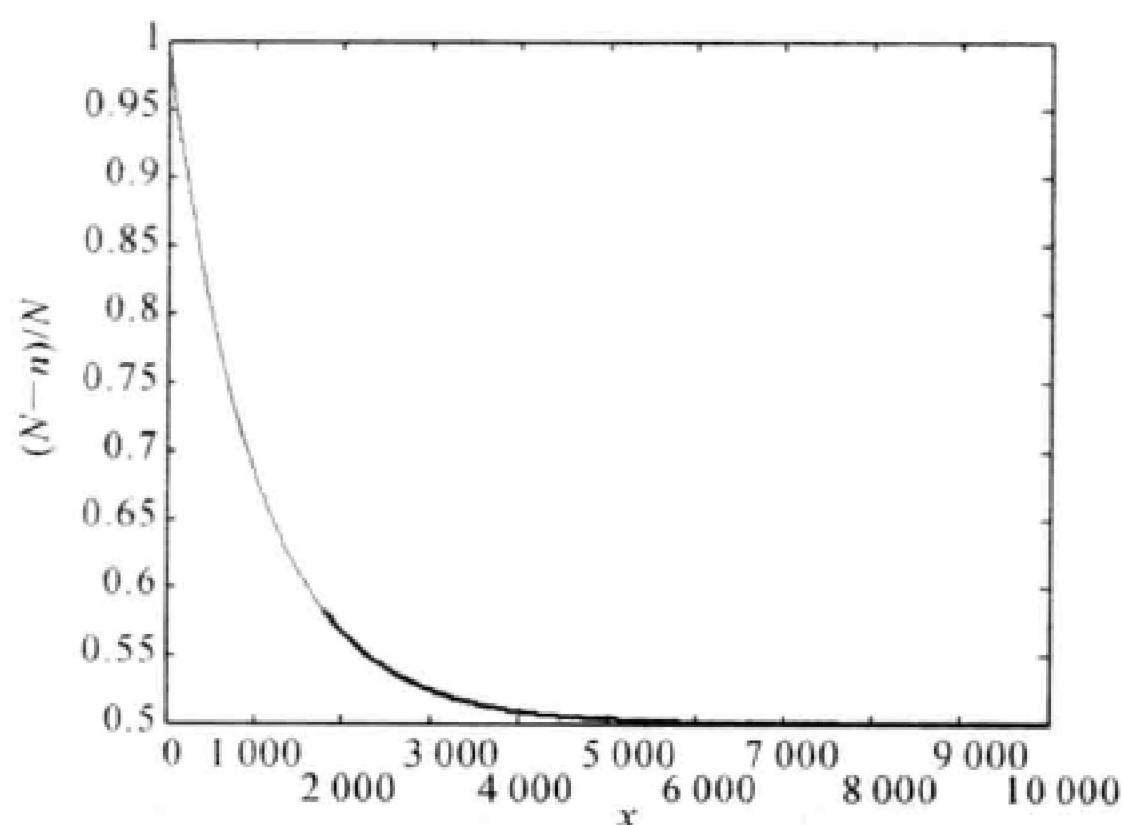


图　9.9

9.4.2　麦克斯韦速率分布规律建立过程的模拟

在一密闭容器内有 N 个相同的理想气体分子，气体系统与外界没有能量交换，即气体系统是孤立系统。无论气体系统的初始状态如何，只要经过足够长的时间，使气体分子之间充分发生碰撞，气体系统将最终达到平衡状态。在平衡状态下，气体分子的速率分布函数是

$$f(v)=4\pi\left(\frac{m}{2\pi kT}\right)^{3/2}v^2\exp\left(-\frac{mv^2}{2kT}\right) \tag{9.33}$$

其中，m 是分子质量，T 是气体温度，$k=1.38\times10^{-23}\,\mathrm{J/K}$ 是玻尔兹曼常量，气体中速率在 $v\sim v+\mathrm{d}v$ 范围的分子数与总分子数的比率为

$$\frac{\mathrm{d}N}{N}=f(v)\,\mathrm{d}v \tag{9.34}$$

这一规律称为麦克斯韦速度分布定律。

这里应用蒙特卡罗方法模拟理想气体分子速率分布的建立过程。首先分析两个气体分子弹性碰撞后速率的变化。设两个分子碰撞前的速度分别是 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$，它们与 x 轴的夹角分别是 α_1 和 α_2，两个分子组成的质点系质心的运动速度是 $\boldsymbol{v}=(\boldsymbol{v}_1+\boldsymbol{v}_2)/2$，如图 9.10 所示。在质心坐标系中，两个分子的运动速度分别是

$$\boldsymbol{w}_1=\boldsymbol{v}_1-\boldsymbol{v}=\frac{1}{2}(\boldsymbol{v}_1-\boldsymbol{v}_2)\,,\quad \boldsymbol{w}_2=\boldsymbol{v}_2-\boldsymbol{v}=\frac{1}{2}(\boldsymbol{v}_2-\boldsymbol{v}_1)=-\boldsymbol{w}_1$$

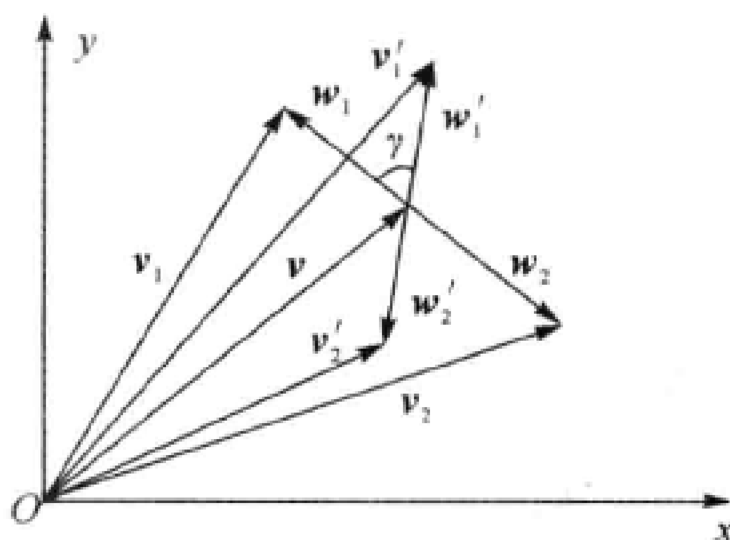


图　9.10

根据动量守恒定律和能量守恒定律，在质心坐标系中，碰撞只能使分子速度 $\boldsymbol{w}_1$ 和 $\boldsymbol{w}_2$ 的方向变化，大小保持不变。用带撇的符号表示碰撞后的物理量。若 $\boldsymbol{w}'_1$ 与 $\boldsymbol{w}_1$ 的夹角为 γ，则

$$\boldsymbol{w}'_1=\boldsymbol{T}\boldsymbol{w}_1\,,\quad \boldsymbol{w}'_2=-\boldsymbol{w}'_1\,,\quad \boldsymbol{T}=\begin{bmatrix}\cos\gamma & \sin\gamma\\ -\sin\gamma & \cos\gamma\end{bmatrix}$$

可以推算出碰撞后两个分子的速率分别由以下两式给出：

$$(v'_1)^2=\frac{1}{2}(v_1^2+v_2^2)+\left[\frac{1}{2}(v_1^2-v_2^2)\cos\gamma+v_1v_2\sin\alpha\sin\gamma\right] \tag{9.35}$$

$$(v'_2)^2=\frac{1}{2}(v_1^2+v_2^2)-\left[\frac{1}{2}(v_1^2-v_2^2)\cos\gamma+v_1v_2\sin\alpha\sin\gamma\right] \tag{9.36}$$

式中，$\alpha=\alpha_1-\alpha_2$。

对气体分子麦克斯韦速率分布规律建立过程的模拟方法如下：

(1) 在全部 N 个分子中随机选择两个分子；

(2) 在 $[0,2\pi]$ 内随机选定碰撞前两个分子速度之间的夹角 α；

(3) 在 $[0,2\pi]$ 内随机选定 $\boldsymbol{w}'_1$ 与 $\boldsymbol{w}_1$ 的夹角为 γ；

(4) 计算碰撞后两个分子的速率 v'_1 和 v'_2；

(5) 多次重复过程(1) ～ (4)；

(6) 使分子之间经过充分碰撞后，计算各速率间隔内的分子数与总分子数的比率，这个比率可以作为分子速率在该间隔内的概率的近似值。

程序(9.6)　模拟气体速率分布的 MATLAB 程序。

```
N = 2000;                                % 总分子数
v = linspace(0, 1000, N);                % 初始时刻气体分子速率在 0 ～ 1 000 之间线性变化
for i = 1:40000                          % 分子之间的碰撞总次数为 40 000
  k = ceil(N * rand(1,2));               % 任意选择两个分子,k 为选定分子的编号向量
  v1a = v(k(1)); v2a = v(k(2));          % 碰撞前两个分子的速率
  a = 2 * pi * rand(1);                  % 在[0,2π] 内随机选定角 α
  r = 2 * pi * rand(1);                  % 在[0,2π] 内随机选定角 γ
  v1b = 0.5 * v1a^2 + 0.5 * v2a^2 + 0.5 * cos(r) * v1a^2 - ...
      0.5 * cos(r) * v2a^2 + v1a * v2a * sin(r) * sin(a);   % 碰撞后第一个分子速率的平方
  v2b = 0.5 * v1a^2 + 0.5 * v2a^2 - 0.5 * cos(r) * v1a^2 + ...
    0.5 * cos(r) * v2a^2 - v1a * v2a * sin(r) * sin(a);     % 碰撞后第二个分子速率的平方
  v(k(1)) = sqrt(v1b); v(k(2)) = sqrt(v2b);                 % 更新两个分子的速率
end
hist(v, 30);                   % 画直方图
hold on
m = 32e-3/6.022e23;            % 分子质量
T = 350;k = 1.381e-23;         % 温度和玻尔兹曼常量
f = 4. * pi. * (m./(2. * pi. * k. * T)).^(3/2). * v.^2. * exp(-m. * v.^2./(2. * k. * T)). * 95000;
                                                                  % 理论值
plot(v, f,' * k');
```

程序运行结果如图 9.11 所示。

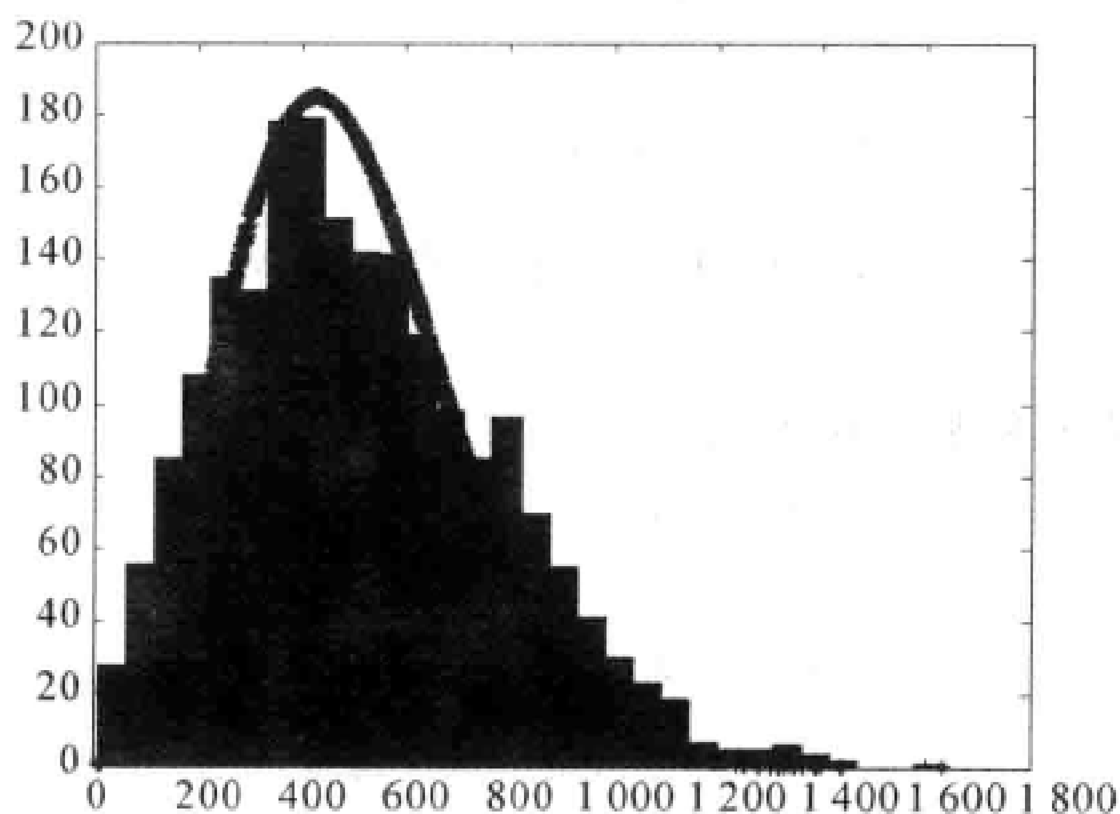


图　9.11

9.5 迭代函数系统

迭代函数系统(IFS)是由巴恩斯利(Michael Barnsley)和斯隆(Alan Sloan)于1987年发明的,它是模拟生物形态最成功的系统之一。迭代函数系统的算法是确定性算法和随机性算法相结合的方法。"确定性"是指迭代规则是确定的,它由一组变换规则构成;"随机性"是指迭代过程是不确定的,每次迭代选用哪一条变换规则是随机的。

9.5.1 塞平斯基三角形

塞平斯基三角形是一个规则分形图形,它是由波兰数学家塞平斯基(Waclaw Sierpinski)于1916年提出的。制作塞平斯基三角形的方法是:① 取一等边三角形,连接各边中点,将三角形分成四个小三角形,然后挖去中间一个小三角形;② 将剩下的三个小三角形按照上面方法同样进行分割,并舍弃位于中间的那个三角形;③ 不断重复分割与舍弃,就能得到塞平斯基三角形,如图9.12所示。

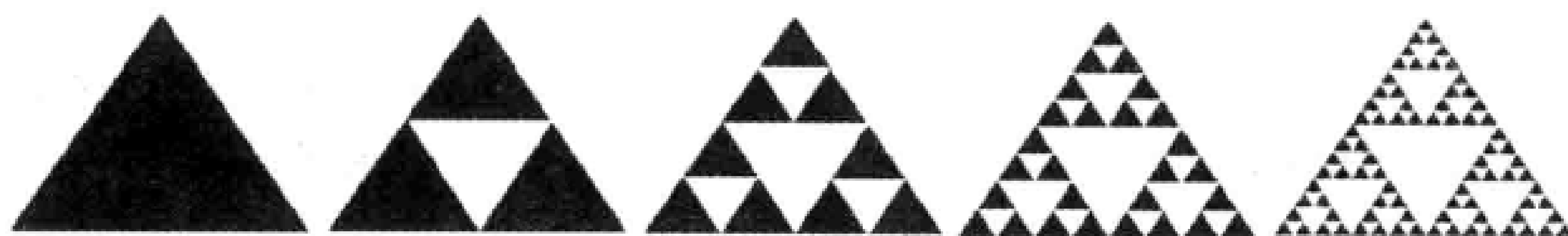


图 9.12

可以用迭代函数系统方法绘制塞平斯基三角形。迭代过程如下:

(1) 在平面上选定三个点 A,B,C 和三种概率 $P_A=1/3, P_B=1/3, P_C=1/3$,以点 A,B,C 作为三角形的三个顶点;

(2) 在三角形内部任选一点作为迭代起点 Z_0,按照以下迭代算法计算新点的位置:

$$Z_{j+1}=\begin{cases}\dfrac{Z_j+A}{2} & (\text{以概率 } P_A)\\ \dfrac{Z_j+B}{2} & (\text{以概率 } P_B)\\ \dfrac{Z_j+C}{2} & (\text{以概率 } P_C)\end{cases}\quad (j=0,1,2,\cdots) \tag{9.37}$$

(3) 以 $Z_j\ (j=0,1,2,\cdots,N)$ 为坐标在平面上画点,当 N 足够大时,平面上就形成清晰的塞平斯基三角形图案。

程序(9.7) 绘制塞平斯基三角形的MATLAB程序。

```
N = 10000;                    % 计算点的总数
i = sqrt(-1);                 % i = √-1
TP = [-1, sqrt(3) * i, 1];    %A,B,C 三点坐标分别是(-1,0),(0,√3),(1,0)
Z = i;                        % 三角形内部选定的迭代起点 Z0 坐标为(0,1)
v = rand(N, 1);               % 生成 N 个在[0,1]内均匀分布的随机数
for j = 2:N
    if v(j) < 1/3
```

```
        Z(j) = 0.5 * (Z(j-1) + TP(1));    % 随机数小于 1/3 时,把 Z_{j-1} 与 A 的中点作为 Z_j 点
    elseif v(j) < 2/3
        Z(j) = 0.5 * (Z(j-1) + TP(2));    % 随机数在[1/3, 2/3]内时,把 Z_{j-1} 与 B 的中点作为 Z_j 点
    else
        Z(j) = 0.5 * (Z(j-1) + TP(3));    % 随机数在[2/3, 1]内时,把 Z_{j-1} 和 C 的中点作为 Z_j 点
    end
end
plot(Z,'.k', 'markersize',4);          % 画 N 个点 Z_j (j = 0,1,2,…,N)
axis equal
```

程序运行结果如图 9.13 所示。

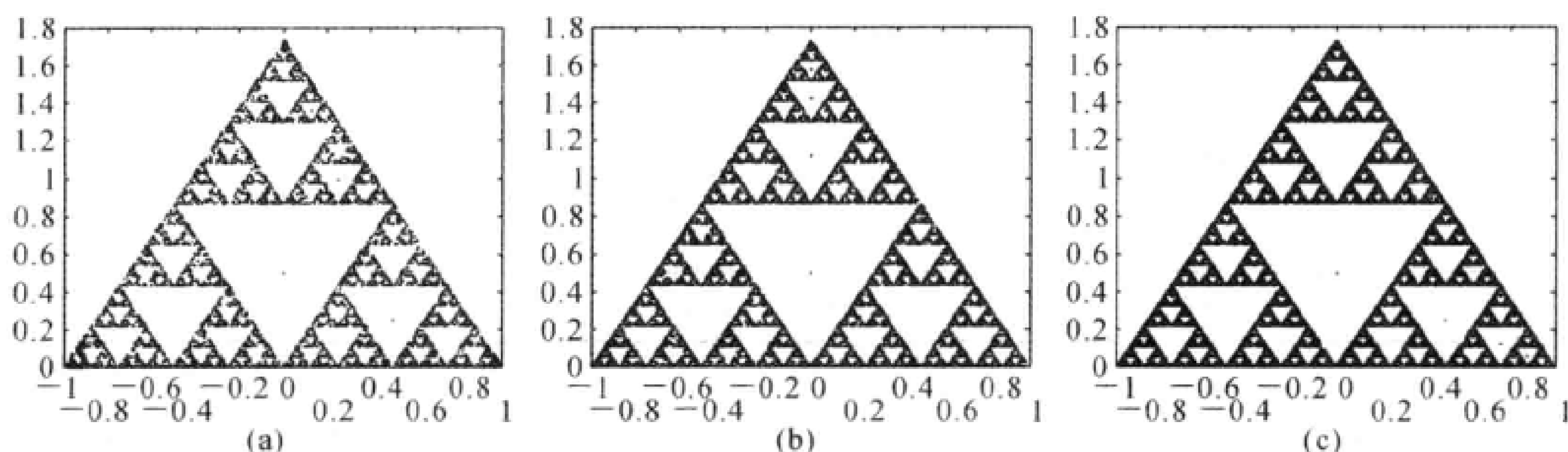


图　9.13

(a) $N = 5\ 000$;　(b) $N = 10\ 000$;　(c) $N = 20\ 000$

迭代算法式(9.37)可以推广到 n 边形的情况。取平面上的点 $Q_k\ (k = 3,4,\cdots,n)$ 为 n 边形的顶点,在 n 边形内任取一点作为迭代起点 Z_0,此时的迭代算法是

$$Z_{j+1} = \begin{cases} Q_1 + d(Z_j - Q_1) & (\text{以概率 } P_1) \\ Q_2 + d(Z_j - Q_2) & (\text{以概率 } P_2) \\ \vdots & \vdots \\ Q_n + d(Z_j - Q_n) & (\text{以概率 } P_n) \end{cases} \quad (j = 0,1,2,\cdots) \tag{9.38}$$

式中,d 为偏离度,取值范围为[0.1, 0.5],但最好取在[0.2, 0.4]之间,以获得较为理想的图形。绘制的四边形和五边形如图 9.14 所示。

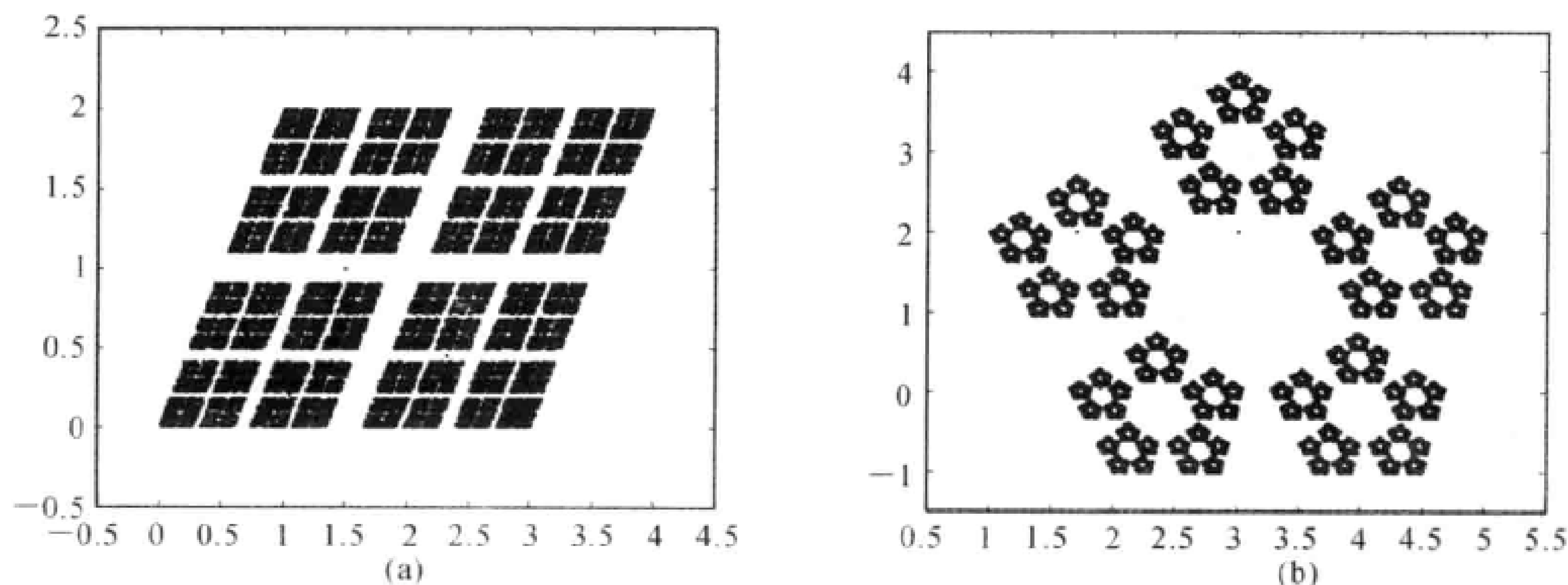


图　9.14

9.5.2 羊齿叶图案

羊齿叶、分形树等都是经典分形图形。绘制羊齿叶图案的迭代过程如下：

(1) 在 xy 平面上选一点 $Z_0(x_0, y_0)$ 作为迭代起点。

(2) 按照以下迭代算法计算新点的坐标：

$$Z_{j+1}=\begin{bmatrix}x_{j+1}\\y_{j+1}\end{bmatrix}=\begin{bmatrix}a_k & b_k\\c_k & d_k\end{bmatrix}\begin{bmatrix}x_j\\y_j\end{bmatrix}+\begin{bmatrix}e_k\\f_k\end{bmatrix}\quad(\text{以概率 } P_k)\quad(k=1,2,3,4;\ j=0,1,2,\cdots)\tag{9.39}$$

上式表示仿射变换，其中系数 $a_k, b_k, c_k, d_k, e_k, f_k$ 及 $P_k(k=1,2,3,4)$ 的值列于表 9.4 中。

表 9.4

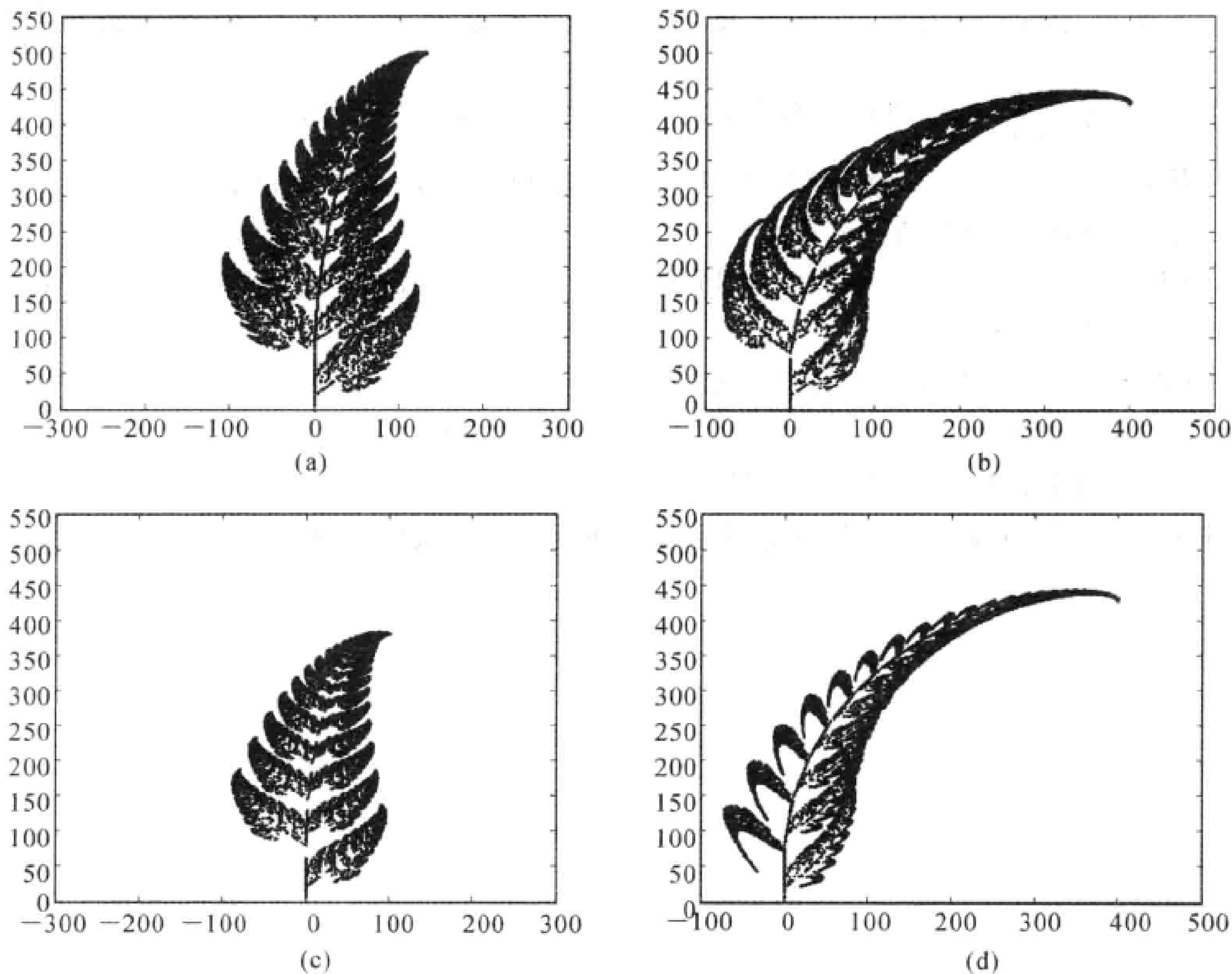
k	a_k	b_k	c_k	d_k	e_k	f_k	P_k
1	0	0	0	0.16	0	0	0.01
2	0.85	0.04	−0.04	0.85	0	80	0.85
3	0.2	−0.26	0.23	0.22	0	80	0.07
4	−0.15	0.28	0.26	0.24	0	20	0.07

(3) 以 $Z_j(x_j, y_j)(j=0,1,2,\cdots,N)$ 为坐标在平面上画点，当 N 足够大时，平面上就会形成完整的图案。

程序(9.8) 绘制羊齿叶图案的 MATLAB 程序。

```
N = 20000;
v = rand(N, 1);
x = 0; y = 0;
for j = 2:N
    if v(j) <= 0.01
        x(j) = 0;
        y(j) = 0.16 * y(j-1);
    elseif v(j) <= 0.86
        x(j) = 0.85 * x(j-1) + 0.04 * y(j-1);
        y(j) = -0.04 * x(j-1) + 0.85 * y(j-1) + 80;
    elseif v(j) <= 0.93
        x(j) = 0.2 * x(j-1) - 0.26 * y(j-1);
        y(j) = 0.23 * x(j-1) + 0.22 * y(j-1) + 80;
    else
        x(j) = -0.15 * x(j-1) + 0.28 * y(j-1);
        y(j) = 0.26 * x(j-1) + 0.24 * y(j-1) + 20;
    end
end
plot(x, y, '.k', 'markersize',4);
axis([-300,300,0,550]);
```

程序运行结果如图 9.15(a) 所示。绘制图 9.15(b) 时，系数 b_2 的值由 0.04 变为 0.14，图形整体向右弯曲；绘制图 9.15(*c*) 时，系数 d_2 的值由 0.85 变为 0.8，图形整体变矮；绘制图 9.15(d) 时，系数 c_3 的值由 0.23 变为 −0.33，图形中左边叶子的弯曲方向发生变化。



(a)　(b)　(c)　(d)

图　9.15

习　　题

1. 编程产生 1 000 个在[0, 1] 内均匀分布的随机数，并对其用参数检验方法进行检验。

2. 用舍选抽样法产生 1 000 个半正态分布的随机数。半正态分布的概率密度函数是

$$f(x)=\sqrt{\frac{2}{\pi}}\exp\left(-\frac{x^2}{2}\right)\quad(0\leqslant x<\infty)$$

3. 分别用投点法和平均值法计算积分 $\int_0^1 \frac{\mathrm{d}x}{1+x^2}$。

4. 半径为 1 的球体被一轴线过球心的半径为 0.5 的圆柱面所截，用蒙特卡罗方法计算球面与圆柱面之间区域的体积。

5. 求解方程 $f(x)=x^3-x-1$ 在区间[1, 1.5] 内的根的近似值 x_0。要求 $|f(x_0)|<10^{-5}$。

6. 以 xy 平面上的三点(0, 0)，(5, 0) 和(3, 3) 为三角形的顶点，用迭代函数法绘制塞平斯基三角形图案。

7.(1) 氢原子 2s 态的电子分布概率密度函数是 $D_2(r)=\frac{r^2}{8a^3}\left(2-\frac{r}{a}\right)^2\exp\left(-\frac{r}{a}\right)$，$D_2$ 的最大值等于 0.14，D_2 的收敛点 $r_2=1.0\text{nm}$；

(2) 氢原子 3s 态的电子分布概率密度函数是 $D_3(r)=\frac{4}{3a^3}\left(\frac{r}{81}\right)^2\left[27-\frac{18r}{a}+2\left(\frac{r}{a}\right)^2\right]^2\exp\left(-\frac{2r}{3a}\right)$，$D_3$ 的最大值等于 0.2，D_3 的收敛点 $r_3=2.0\text{nm}$。

这里，$a=5.29\times10^{-2}\text{nm}$。试模拟氢原子 2s 态和 3s 态的电子云分布。

8. 塞平斯基地毯的构造方法如下：

(1) 取一正方形，把它分割成大小相同的 9 个小正方形，舍弃中心处的小正方形，保留周边的 8 个小正方形；

(2) 把 8 个小正方形都分割成大小相同的 9 个更小的正方形，舍弃每个小正方形中心处的更小的正方形，保留每个小正方形周边的 8 个更小的正方形；

(3) 如此反复分割、舍弃与保留，剩下的图形就是塞平斯基地毯。

将这种构造法则推广到空间就能得到塞平斯基海绵。编程绘制塞平斯基地毯图案。

参 考 文 献

[1] 郭立新，李江挺，韩旭彪. 计算物理学. 西安：西安电子科技大学出版社，2009.

[2] 彭芳麟. 计算物理基础. 北京：高等教育出版社，2010.

[3] Nicholas J Giordano，Hisao Nakanishi. 计算物理. 2 版. 北京：清华大学出版社，2007.

[4] 刘金远，段萍，鄂鹏. 计算物理学. 北京：科学出版社，2012.

[5] 董键. Mathematica 与大学物理计算. 北京：清华大学出版社，2010.

[6] John H Mathews，Kurtis D Fink. 数值方法（MATLAB 版）. 3 版. 陈渝，周璐，钱方，译. 北京：电子工业出版社，2002.

[7] 吕同富，康兆敏，方秀男. 数值计算方法. 北京：清华大学出版社，2008.

[8] 胡兵，李清朗. 现代科学工程计算基础. 成都：四川大学出版社，2003.

[9] 沈剑华. 数值计算基础. 上海：同济大学出版社，2004.

[10] 贺俐，陈桂兴. 计算方法. 武汉：武汉大学出版社，1998.

[11] 蔡大用. 数值分析与实验学习指导. 北京：清华大学出版社，2001.

[12] 王沫然. MATLAB 5. X 与科学计算. 北京：清华大学出版社，2000.